DOCUMENT RESUME

ED 264 842 IR 011 926

TITLE	The National LOGO Exchange Newsletter, Volumes 1-3 (September 1982-May 1985) with Topical Index.
INSTITUTION	National LOGO Exchange, Charlottesville, VA.
PUB DATE	85
NOTE	375p.; Topical Index was compiled by Regina Sapona.
AVAILABLE FROM	Editor, National LOGO Exchange, PO Box 5341, Charlottesville, VA 22905.
PUB TYPE	Collected Works - Serials (022) -- Guides - Classroom Use - Guides (For Teachers) (052) -- Reports - Descriptive (141)
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	Book Reviews; *Computer Graphics; Elementary Secondary Education; Indexes; *Learning Activities; *Merchandise Information; Newsletters; Program Descriptions; *Programing; *Programing Languages; *Teaching Methods
IDENTIFIERS	*LOGO Programing Language

ABSTRACT

This document consists of 27 issues of the National LOGO Exchange Newsletter (NLX) together with cumulative indexes which provide five access points to information in the newsletters: (1) an author index; (2) a book review index; (3) a product and information index; (4) a column index; and (5) a subject index. The alphabetical author index lists the author's name, the month and year of the newsletter in which the article appears, and the page number. Book reviews are listed by the first author of the book reviewed and include reviews for books mentioned in columns as well as more in-depth reviews. The following columns are listed alphabetically by title with the volume numbers and dates of issues in which they appeared: From the Editor; Listful Thinking; Logo Disserts--Dissertations Dealing with Logo; MAE 1000--Logo in Japan; Microworlds; NLXionary: A Lectionary of Selected Logo Readings; NLXual Challenges; Q and A; Teacher to Teacher; Tipps for Teachers; and Turtle Tips. The authors are listed for each column. Both the product and information index and the subject index provide newsletter dates and volume issues for descriptions or reviews of specific Logo-related terms, and the subject index also lists authors, operational commands, and book titles. (JB)

****
*   Reproductions supplied by EDRS are the best that can be made   *
*   from the original document.   *
****

ED264842

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

● Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

The National LOGO Exchange

September 1982 - May 1985

Volumes 1-3, Numbers 1-9

with

Topical Index

IR011926

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY
Tom Lough
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

BEST COPY AVAILABLE

ISSN 0734-1717

The National LOGO Exchange

Volume 1 Number 1 FORWARD 100! September 1982

Mindstorms — Don't Miss It

If there is to be a classic book about LOGO, it is sure to be Mindstorms by Seymour Papert. It is practically impossible to know anything about LOGO without having heard something about Mindstorms, since Dr. Papert was one of LOGO's founding fathers some twelve years ago.

If you have not yet read this book, don't miss it. The essence of the LOGO learning philosophy is contained therein, written in an easy, straightforward style. But that is not all that awaits you.

Papert develops many powerful ideas which are of importance and concern in today's education systems. Most deal directly with the role of computers in the classrooms.

His main thrust seems to be in the direction of "natural learning," a philosophy he gained from direct study with Piaget. He views LOGO as an ideal vehicle to provide the children with a learning environment in which they can explore, experiment, and play, using the computer and its capabilities in self-directed ways.

He also looks to the future, and sees the need for a revolution in education as we know it today! The QWERTY phenomenon is his name given to any process which is so deeply established that, regardless of any suggested improvements or developments, no changes are permitted. "We have always done it this way," is the motto of the QWERTY-ite. Does this idea sound familiar? Perhaps you will find this part of Mindstorms extremely stimulating!

If you are looking for LOGO lesson plans, you will not find them in this book. But, you will find several concete ideas and specific philosophical concepts upon which you can easily base your LOGO plans, activities, and curricula. And you will gain an appreciation for the incredible potential of LOGO, the next giant step in education!

Mindstorms: Children, Computers, and Powerful Ideas, Seymour Papert, Basic Books, New York, 1980.

LOGO Reinforces Geometric TABS Skills

by
Linda Nix

The computer was a new addition to our school this past year, and, as the fifth grade teacher, I was anxious to use it as much as possible as an extension of my regular teaching strategies. In particular, I wanted to use it to reinforce required TABS skills. I found LOGO to be an exciting and effective way to develop recognition of geometric shapes.

My technique was for students to recreate geometric shapes by using LOGO and to record their programs in a notebook for future use by other students. Since LOGO was new to both my students and me, I started slowly with a teacher-directed activity demonstrating computer commands in basic LOGO for the turtle mode. The students sat on the floor in front of the computer and took turns giving directions to the computer, which I typed in, to draw a particular shape.

Several students, because of their scounting activities, quickly saw the relationship between the degrees on a compass and the size of the turns required to make the correct angles.

After drawing a square, a rectangle, and a triangle, we drew the outline of a house. It was exciting to observe the group interaction that took place. Even the shyest child became involved.

The students were then paired with partners and given the assignment to write directions for the computer for other geometric shapes. They worked at their desks using paper and pencil. Some used a compass to help them decide on the size turns to have the computer make.

During this time, it was interesting to watch the decision making process that was taking place and the student interaction. Many students worked together beautifully. For others, it became a test of patience and self-control.

continued on page 4

IR011926

The National LOGO Exchange, copyright 1982 Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly September through May, $25 per year, $45 for two years, mailed FIRST CLASS from Charlottesville, VA. Add $5 per year for addresses outside the United States. Single copy price $3. Opinions expressed by the authors are not necessarily those of The National LOGO Exchange. Address all editorial correspondence to the editor at The National LOGO Exchange, Post Office Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor......................Tom Lough

WELCOME!

Welcome to The National LOGO Exchange, a monthly newsletter for LOGO teachers and parents! We hope that you will be able to make use of the tips and techniques which are presented to you here.

In each issue, we will have columns to help you develop LOGO plans and activities. Book reviews and critiques will be included, as well as comments on LOGO products we have reviewed. Important features will be articles written by classroom teachers and parents, reporting on their LOGO activities and discoveries.

We also hope that you will be willing to share your experiences and lessons in LOGO with your fellow readers. The future for LOGO is so exciting! Here is an innovation which is capable of changing education as we know it today. How its use develops will determine its effectiveness, to a large extent. You are in a position to influence that development by reporting your LOGO successes to others so their children can also benefit. Why not write your LOGO article today?

Here at The National LOGO Exchange, our motto is FORWARD 100! This reflects our enthusiasm toward LOGO and its role in education. We are very excited about the potential represented by LOGO. We want to press on in a FORWARD direction with our efforts to bring LOGO to a position from which it can influence our country's children. We want to give 100% support to you, the LOGO teachers and parents, as you work with LOGO in the classrooms and homes. Please feel free to adopt this motto as your own, for LOGO, and for life!

FORWARD 100!

Tom Lough
Editor

TIPPS for TEACHERS

by
Steve Tipps

Distance and Direction

The first four primitive commands which are introduced to the new LOGO programmer are FORWARD, BACK, RIGHT, and LEFT. With these commmands, the turtle can be directed to move in many directions and draw an infinite number of figures. However, the four commands also are codes for the first two concepts which are embedded in the turtle's microworld: distance and direction.

Distance (how far) and direction (which way) are two different ideas which children and adults need to recognize as separate ideas if they are to achieve an understanding of the geometric and arithmetic rules which govern the turtle. These concepts need to be introduced at the body and toy level before the computer and screen turtle are used. A number of simple direction games common in preschool and primary classrooms can be adapted for use.

Body games. "Simon Says" and "Mother, May I?" are the two most common variations of body direction games. Simon says, "Take two giant steps," is easily changed to "Take two turtle steps." I usually emphasize that turtles take rather small steps, so that even 10 turtle steps would probably not equal one giant step. Children can also walk off the dimensions of the room or the hallway using turtle steps. This activity is also used as an introduction to measurement and units.

The emphasis has been on distance and, in particular, the command FORWARD. However, BACK commands can also be used in "Mother, May I?" and in measuring distance. Many times, children will want to turn around and step FORWARD when given the instruction BACK. The teacher would emphasize that, for both FORWARD and BACK instructions, the child's face is pointing the same direction. This provides an opportunity to introduce turning or direction as a separate concept.

continued on page 3

Tipps for Teachers continued

In the kindergarten and primary grades, teachers usually teach left and right as labels for hands and feet. LEFT and RIGHT commands can be incorporated directly into the body games. "Turn 10 turtle turns to the LEFT," or "Make a giant turn to the RIGHT," gives reasons to know and ways to use directions. Children can argue about and define the size of a giant turn. Coming to a class agreement about the size is in keeping with the child centered philosophy of LOGO.

Toy games. When the concepts of distance and direction are established, they can be used in a variety of ways. "Control Tower" is one activity which my students always liked. One student would leave the room while the others would rearrange the desks into a maze and a landing strip. The task of the Air Traffic Controller or Committee was to guide the blindfolded student by giving commands of FORWARD, RIGHT, or LEFT. (BACK is used with a helicopter, but students will not allow it with an airplane!) The object of the blindfolded pilot, of course, is to follow the directions carefully and quickly to bring the jet down on the runway. Whether there was actually a toy airplane or not, children enjoyed the mock situation, and would invent a variety of dramatic reasons for the pilot and plane being disabled.

Variations with automobiles, submarines, and helicopters are possible. Routes and roads can be made out of blocks. Maps of the school and routes from home to school or to the grocery store can be developed. "Mystery Places" for third and fourth grades are a good class puzzle. One or two children can describe a route from the school to an unknown destination by counting blocks and turns. Similar activities should be part of mapping units in social studies, science, or math for the primary grades. In each case, the differences between distance and direction are the primary conceptual learnings.

If the teacher has access to a floor turtle (available from Terrapin Inc., 678 Massachusetts Avenue, Cambridge, MA 02139), the direct transfer to the computer turtle is apparent. However, if no floor turtle is available, the intermediate steps with objects should not be skepped. On a large sheet of manila or graph paper, students can draw figures and designs which are dictated to them. Different magic markers can be used. While this is a crude simulation of a floor turtle, the drawing is a good representation of what will be done with the screen turtle.

continued

Screen games. Before students get to the computer and type in FORWARD, BACK, RIGHT, or LEFT, I am convinced that they need adequate time to explore the ideas of distance and direction at the more concrete levels of the body and the toy. The first problem which students (children or adults) encounter is recognizing that a RIGHT or LEFT command does not cause the turtle to draw a line. The amount of time which is spent being mystified seems to be related to the ability to convert the commands into a more concrete level and see that direction commands never cause any movement. The amount of time which needs to be spent at the more concrete levels may vary from individual to individual and age to age, but it is a foundation for the thinking and debugging process of LOGO. Without the foundation and process, I have observed learners swear that the only place where the turtle can draw a square is in the center of the screen! Full control of the turtle is based on a full understanding of the concepts of direction and distance.

A number of screen games can be played or invented to increase awareness and facility of control. One game is "Hit My Finger," in which one child puts a finger on the screen and the other tries to guide the turtle to it. A lot of experimentation will go on as the pointer tries to find more difficult places to put a finger, and the seeker tries to cut down the number of separate moves needed to touch. The exact rules of success can be left to the ingenuity of the children (who, by the way, will usually set increasingly exact standards!). A variation is to put colored adhesive dots on the screen for children to reach. A declaration of "I can hit the red dot in four moves," may be the beginning of a bidding process which children soon find has a limit of two moves, except for certain places on the screen.

The teacher can change the game by establishing a procedure called SPIN (listed elsewhere in this issue) which randomly sets the direction of the turtle at HOME. (This brings up the question of whether or not a teacher should ever preprogram anything for children. This important issue has many thorns which will be dealt with in later columns. In this case, I feel that, since the purpose of the game is to increase understanding of distance and direction, this small preprogram serves that instructional purpose.) A more complex distance and direction procedure is suggested in the Solomon

continued on page 8

TABS continued

I then posted a schedule of times the computer would be available to each group to test their programs. The students could sign up for one class period, before or after school times, or during recess.

During this entire testing of programs, regular math assignments were continued. After the programs were fully tested, the students recorded them in a notebook for future reference. Several of the students wrote programs containing more than one geometric shape. Later in the year, students shared their programs with other grade levels experimenting with the LOGO language.

This coming school year, I plan to expand this activity a step further by doing the exercise earlier and saving the students' programs on a diskette. Later, the students can run the programs as an exercise to review recognition of geometric shapes prior to testing.

As the year progressed, it was exciting to watch the students experiment and discover interesting ways of making the computer do what they wanted, using turtle and sprite modes. They developed an awareness and appreciation of what was involved in making the computer games they love to play. It reinforced my opinion that LOGO can be an exciting and valid addition to an elementary math computer curriculum.

Linda Nix is a fifth grade teacher at Forest North Elementary School in the Round Rock Independent School District, Austin, Texas.

SPIN

Here's a procedure to set the turtle to a random heading no matter where it is located. This could be used in many introductory exercises, as suggested in the Tipps For Teachers column of this issue. I'm confident that you could devise many more creative uses for SPIN.

If you or your students work with SPIN and achieve some interesting results, write them up for publication! It is through our sharing that our children benefit.

For the Apple:

```
     TO SPIN
        SETHEADING RANDOM 360
        END
```

continued on page 8

You're Invited!

The avalanche is beginning! LOGO is on the way ... and it has the potential to revolutionize education as we know it!

But, the teachers across the country need your help. They are beginning to experience challenges which you already have met and fears which you already have faced down. Would you like an opportunity to be of service to untold numbers of fellow LOGO teachers and their students? If so, then consider writing a short article about an incident from your LOGO experience for The National LOGO Exchange.

This monthly newsletter presents articles written by LOGO teachers for LOGO teachers. We are looking for short (500 words or less), snappy, practical pieces about specific LOGO incidents from which you learned about yourself and how you were teaching. Do you have a neat technique or method for encouraging children to explore a certain concept? What was your very first day with LOGO and children like? "If I had it to do all over again, I would be sure to ... " What are your favorite programs developed by your students? How did you help them discover the difference between :variable and "variable?

You could double the number of such questions in less then 15 seconds, I'm sure. Why not take a few minutes to write about one or two, or more?

And please do not limit yourself necessarily to those things which worked. The beginning LOGO teachers need to know what things might <u>not</u> work as well!

Parents, we need to hear from you, too! What are you experiencing with LOGO? What are your concerns? When you observe how your child is thinking when working with LOGO, what are your thoughts? Do you have any ideas or suggestions which teachers might find helpful for their LOGO work?

Mail your articles to The National LOGO Exchange, Box 5341, Charlottesville, VA 22905.

We thank you and look forward to hearing from you. Your fellow LOGO teachers thank you and look forward to reading about you. The children thank you and look forward to benefitting from you.

FORWARD 100!

MICROWORLDS

by

Glen Bull

Design of a Language

Consider the following:

- An educator suggests that a subset of BASIC (eight - command BASIC) may be a more effective introduction to programming for children than LOGO.

- A salesman in a computer store tells a customer that a version of Pilot with turtle graphics is "LOGO."

- A graphics language with limited test-handling capabilities is described as a new implementation of "LOGO" in an article in a national publication.

The three occurrences described above have a common element in that they all betray fundamental misconceptions of the true nature and intent of LOGO:

- LOGO is better suited to educational use than BASIC because of the structure of the language, rather than because there are fewer commands to learn.

- LOGO is more than turtle graphics. Turtle graphics is a highly visible feature of LOGO which has influenced development of graphics capabilities in a number of other languages, including dialects of Pascal, Pilot, and Smalltalk. Even so, powerful operators for manipulation of language are also an integral part of LOGO.

Papert argues that failure of educators to consider the design of educational languages will inevitably result in inappropriate structures, by default. The danger seen is not that the wrong choices will be made. Rather, the problem lies in the shaping of educational philosophies through lack of awareness that there are choices to be made

continued

The structure and characteristics of LOGO are not mystical, ineffable qualities beyond discussion. They are straightforward design specifications subject to examination. To do otherwise is to act from blind faith or religious fervor.

The most important characteristics which might be desired in a language intended for use with children can be summed up in three words: interactive, procedural, and extensible. BASIC was the first language designed to teach programming which was used interactively on a widespread basis. The importance of this advance would be difficult to overestimate. When centralized computing resources are expensive and scarce, they can be optimized by making people wait in a queue for processing time. Under those conditions, it may be minutes or hours before the results of a program are seen.

On the other hand, direct access to the computer and immediate results undeniably have a strong effect on learning. This philosophy suggests that a person's time is more valuable than a computer's. This may seem obvious now. At the time BASIC was developed, it was a suspect, controversial idea. It would be over a decade before the term "personal computer" was needed, in the sense of a computer inexpensive enough for exclusive use by a single person.

The wide-spread use of BASIC made it evident that it was highly desirable for non-professional users of computers to employ the computer in an interactive fashion. After a time, it also became apparent that BASIC was not the ultimate, perfect language. In fact, computer scientists concluded that the structure of languages such as BASIC fostered development of poor programming habits and error-ridden programs.

There are a number of reasons this is so. An important one is the fact that BASIC is not procedural, in the true sense of the word. Its closest approximation to a procedure, the subroutine, falls short in several respects. In a programming language, a procedure is an independent description of a process which performs an action. Once developed, this description can be treated as a black box by the remainder of the program. The mechanics of how the goal is accomplished inside the black box can be disregarded.

continued on page 6

Microworlds continued

The best-known procedural language developed for teaching programming is Pascal. It has almost universally been adopted for teaching programming by computer science departments. Pascal was developed for college students. Hence, a considerable knowledge of computer science is required to write even a short program in Pascal. Also, Pascal programs cannot be run as soon as they are typed, but must be compiled first. The benefit of Pascal is that it does encourage good programming practice, even when the user moves on to other languages.

LOGO is both interactive and procedural. A procedure can be run as soon as it is typed. A design goal of LOGO was an educational language with no threshold and no ceiling. In a certain sense this goal has been met, since LOGO has been successfully used by children in elementary school and by physics students at MIT. In passing, it might be mentioned that the common impression that LOGO is only of interest to children is incorrect. LOGO has facilities for developing routines in assembly language, as well as list processing capabilities found in few languages.

In LOGO, a procedure is invoked by naming. As a consequence, the langauge is extensible. This means that the user can develop new commands in LOGO, and, in effect, write a new language tailored to a particular need or application. This capability is only available in a few other languages, such as Forth. For example, LOGO does not provide a FOR ... TO control structure similar to those found in BASIC and Pascal. The philosophical emphasis of LOGO seems to be on recursive rather than iterative control structures. However, a FOR ... TO command can easily be developed in LOGO, if there is a need for this structure.

In its current implementations LOGO is, unsurprisingly, not a perfect language. The intended application of a language heavily influences its efficiency. LOGO is not extremely fast. Because it is a sophisticated language, it requires more memory than languages such as BASIC. In a classroom environment speed of execution and efficient use of memory are usually not the most crucial considerations.

The price of silicon computer chips will probably fall rapidly, and their capabilities will increase. As this occurs, limitations of speed and memory will become less of a factor. However, language characteristics which are artifacts of technologic limitations will continue to be perpetuated unless thought is given to desirable traits. For example, usually the absence of procedures

continued on page 8

Tell Congress:

We Want Computers!

Those of you who are participating in the computer revolution will have a particularly important reason for making your presence known. There are two actions in Congress which need your support: The Technology Education Act of 1982 (HR 5573 and S2281) and The Family Opportunity Act (HR 6397).

The Technology Education Act of 1982 is of special importance to schools. It provides incentives for the major computer manufacturers to donate a state-of-the-art computer to individual schools. With only a few hundred thousand computers in schools to date, this bill, if passed into law, will provide a veritable avalanche of powerful computers for many schools, particularly for those less gifted financially (are there any otherwise?). But it is up to the legislators, our senators and congressmen, to vote on the measure. If we will take the few minutes necessary to write a note outlining our support of the bill, we will be doing our part to help make it work.

If you are a teacher, why not check with your principal to see if you could send in your letter on your school's stationery? How about getting the parents' organization involved? They could work on gathering signatures on petitions. Do your students want more computers in their schools? Let them participate in the democratic process. Writing a letter to Congress makes a dynamite social studies and English assignment!

The other bill, The Family Opportunity Act, provides for a tax credit to a family for the purchase of a computer system and software. This will encourage families who can afford computers, and who have just been "waiting to see what will happen," to take the plunge now. This will put computers within the reach of many children during non-school hours. Think of the growth opportunities this might provide!

In the background of all these remarks, of course, are tremendous implications for LOGO. Just think of all the gateways these two bills could open for LOGO activities if they become law! Our education system could experience a growth like nothing which has been experienced to date. Imagine what it would do for our children if nearly all of them would have the opportunity to grow up learning with LOGO! Don't you feel the children are ready for it? Don't they deserve it?

LOGO Your Boat

For those of you involved with physics, here are a few things LOGO to think about when dealing with a particular class of vector problems.

Remember the situation of a boat and a stream? In general, the problem is to figure out how to paddle the boat across the stream and arrive at a particular point on the opposite bank, knowing the speed of the stream and the boat.

Most physics texts encourage the student to figure out the constant angle at which to head the boat upstream, so that the vector sum of the speed of the boat and that of the stream will be a vector which points directly across the stream to the desired destination.

This is a correct approach, as far as it goes. However, as those of you who actually have paddled a boat or canoe know, keeping a constant angle with respect to the current while vigorously paddling is difficult to do in practice.

Rather, the easiest thing to do is to keep paddling the boat toward a certain distant point. If that point is your ultimate destination, you might eventually get there.

LOGO can be used readily to study this situation. Depending on the speed of the stream and the boat, it is possible to come extremely close to your destination. However, for choices much different from a particular range of combinations, you might end up quite a distance downstream, depending on your definition of "arriving at the other side."

The LOGO listing is this article is presented to you for exploration. Feel free to vary the parameters, the dimensions, and the test for arrival. The shapes of the various paths of the boat might hold some surprises for you. If you are a teacher, you might want to make the WATERCROSS procedure available to your students for informal research.

Initally, a stream is set up on the screen. The left and right banks are shown as vertical lines, and the "perfect path" to the destination is shown as the horizontal straight line. The procedure RIVER draws all of this and then places the turtle on the left bank facing the POST on the opposite side of the stream.

When the CROSS procedure begins, the boat (turtle) is paddled FORWARD for one increment of time at the BOATSPEED. Then the stream takes a turn, pushing the boat downstream (towards the top of the screen) at the STREAMSPEED for one increment of time. The heading of the boat is then reset toward the destination (SETHEADING TOWARDS :POST). Testing of arrival is performed next, with the initial criterion being to be within 2 units of the opposite shore. Then, the CROSS procedure calls itself again.

continued

When the test determines that "you have arrived," the distance you are located downstream of the POST is printed at the bottom of the screen, along with a recap of the boat and stream speed.

As mentioned before, you can get some interesting results by varying the test criterion. Instead of 2, for example, try smaller and larger numbers.

The listing below is written for the LCSI Apple LOGO. Changing over to other versions should not be too much of a problem, I hope.

```
    TO WATERCROSS :BOATSPEED :STREAM-
SPEED
      MAKE "POST [ 130 -50 ]
      RIVER
      CROSS
      (PRINT [ DOWNSTREAM DISTANCE IS
] (YCOR + 50))
      (PRINT  [ FOR  BOAT SPEED OF ]
:BOATSPEED)
      (PRINT [  AND STREAM SPEED OF ]
:STREAMSPEED)
      END


    TO RIVER
      HT CS PU BK 50 RT 90 PD
      REPEAT 2 [ BANK ]
      SETPOS [ -130 -50 ]
      SETHEADING TOWARDS :POST
      ST
      END


    TO BANK
      FD 130 LT 90 FD 240
      LT 90 FD 130
      END


    TO CROSS
      FD :BOATSPEED
      SETHEADING 0 FD :STREAMSPEED
      SETHEADING TOWARDS :POST
      IF (130 - XCOR < 2) [ STOP ]
      CROSS
      END
```

If you work with WATERCROSS and wish to share your findings with others, please feel free to write up a brief report or note and send it to us. We would be interested especially in knowing about your extensions or other applications of the procedures or the concepts.

Many times, there is a difference between the "perfect theoretical solution" to a problem and the actual solution which works in the real world. WATERCROSS is one small example of how LOGO can help us to explore differences such as this quickly and in an interesting way.

FORWARD 100!

Tipps for Teachers continued

Apple LOGO manual which randomly positions a target box on the screen and randomly sets the location and direction of the turtle. I might also use this TARGET procedure for early learners of LOGO, but I think its most appropriate use would be as a later project for students to create without the manual.

By the time the learners have explored and mastered distance and direction, they are probably ready to describe the size of the turtle screen across the top, bottom, and the middle, and along both sides and the diagonal. They should be able to guess distances fairly well in turtle steps. The understanding of turtle turns and their size generally lags behind the understanding of distance. I do not teach "degrees" as a prerequisite description of direction ... however, I do not hide the term, either. "Turtle Turns" does very well for younger children. "Square corner" is as explanatory as "right angle" or "90 degrees," although choice of words changes with the learner. A great advantage of LOGO is that prior knowledge of terminology does not limit even the youngest learner from creating. However, incomplete or inaccurate concrete understanding of concepts of distance and direction will inhibit the more sophisticated "turtling." Without adequate preparation, the power of programming with LOGO will be diminished.

Steve Tipps is a professor at the University of Virginia, and has presented LOGO workshops for teachers throughout the eastern United States.

SPIN continued

For the TI 99/4 and 99/4A:

```
     TO SPIN
        MAKE "HUNDREDS RANDOM
        TEST :HUNDREDS<4
        IFF SPIN
        MAKE "TENS RANDOM
        TEST :HUNDREDS=3
        IFT TEST :TENS>5
        IFT SPIN
        MAKE "ONES RANDOM
        MAKE "DEGREES (:HUNDREDS *
100) + (:TENS * 10) + :ONES
        SETHEADING :DEGREES
        END
```

An additional application of the SPIN procedure might be in scouting activities. As mentioned in Linda Nix's page one article in this issue, many of her students made rapid LOGO progress as a result of previous scouting experience with compasses. The SPIN procedure could be used by scouts to practice estimating headings by observing the orientation of the turtle on the screen. Their guesses could be confirmed by PRINT :HEADING.

continued

Microworlds continued

which can resursively call themselves is the result of memory and speed limitations, rather than a conscious decision that a language would be improved by the absence of this capability. Therefore, if these features are desirable, a continuing review is necessary to ensure that the omission of the feature does not continue after the technologic constraints are removed.

LOGO was developed over a number of years in an environment in which cost was not a constraining factor. The only stated limitation was usability of a characteristic in a learning environment. This process was begun over a decade ago. Now some of the benefits of this effort are being reaped as many of these characteristics are becoming possible in microcomputer-based implementations.

Three important aspects of LOGO are embodied in the fact that it is interactive, procedural, and extensible. This by no means exhausts the list of capabilities which define LOGO. Other aspects such as encouragement of meaningful procedure and variable names by removing restrictions on their length, the presence of a resident screen editor, and effective means of passing parameters between procedures all contribute to the underlying quality of LOGO.

Some components of LOGO can be altered or deleted without seriously degrading its fundamental structure. Some implementations of LOGO have arrays, sprites, and floating-point arithmetic, while others do not. Depending on the intended use, these omissions may not seriously cripple the language. However, beyond a point it becomes questionable whether a language can seriously be described as "LOGO," given the original intent of the language. Effective utilization of LOGO will ultimately depend on teachers who understand its characteristics, and the reasons for their inclusion in the language.

Glen Bull is a professor at the University of Virginia, and has taught LOGO courses at the graduate and undergraduate level.

Also, if you are observant, you will notice that the SPIN procedure on the TI computers will orient the turtle to certain "temporary" headings on occasion. The explanation for this is dependent upon a full understanding of the LOGO iterative process. Can you figure it out?

FORWARD 100!

ISSN 0734-1717

The National LOGO Exchange

Volume 1 Number 2 FORWARD 100! October 1982

Turtle Geometry

Don't Wait for the Movie

There are those who regard turtle graphics as little more than "kid stuff," with appeal limited only to elementary school children, and potential limited to mastering a few simple turtle commmands.

People in the above category are not to be "put down" for their gross ignorance. They simply have not yet had the opportunity to read "Turtle Geometry," a textbook by Harold Abelson and Andrea diSessa. That this book is one of substance is indicated in the forward: it is but one volume in the MIT Press Series in Artificial Intelligence.

The next clue comes from the table of contents. It starts off easily enough with turtle graphics and a treatment of the familiar POLY procedure. The reader is then led into looping (closed curve) programs, and is introduced to the powerful idea of feedback. Building upon this, Abelson and diSessa take up the processes of growth and recursion.

So far, so good. But next, the "heavy part" begins. And if you are still thinking at the 3rd grade level at this point, try this list of topics for size!

Here come vector analysis, turtle state, three-dimensional turtle space, rotations, projections, topology, deformations, turtle spherical geometry, curvature, turtle on a cube (!), piecewise flat surfaces, Euler characteristics, curved spacetime and (are you ready for this?) Einstein's Theory of General Relativity!

It is beginning to sound like a college geometry or physics course, and we haven't even made it to the first chapter! But there is no reason to be apprehensive.

In spite of the depth of the subject matter, the authors have described the ideas and examples so clearly that a person with no more than an "average" college education is able to follow and understand.

continued on page 6

A Parent Looks at LOGO

by

Sandy Towberman

Having observed my nine-year-old working with the LOGO turtle, I am convinced that LOGO is a way to create something close to the ideal learning situation, provided that the teachers take advantage of its benefits.

I would like to share my observations with you. Often, it seems, when it comes time to make a decision about innovative ideas in education (and LOGO is most certainly one!), parents are rarely consulted. Or, perhaps worse, their response may be assumed. I want you to know how I feel about LOGO for my child.

I still have much to learn about LOGO. But, as a parent, I can immediately identify three areas with major educational benefits for children.

EXPLORATION

One of the keys to optimum learning is the student proceeding at a self-paced rate, learning something thoroughly before moving on to something else.

I feel that LOGO meets and beats this criterion: it harnesses a child's natural curiosity, allowing him to explore new frontiers. It enables him to discover basic principles by making his own mistakes and leaning from them (a skill of living). He can repeat things as often as necessary to "double-check."

Through working with LOGO, a child learns to integrate one principle with another, not merely learning the principles, but the implications of each one and when to apply it. Such "experimental learning" is invaluable.

A related by-product is that a child can develop a keen understanding of research, and of the scientific method of studying cause and effect by changing only one variable while keeping all others constant.

continued on page 2

The National LOGO Exchange, copyright 1982 Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly September through May, $25 per year, $45 for two years, mailed FIRST CLASS from Charlottesville, VA. Add $5 per year for addresses outside the United States. Single copy price $3. Opinions expressed by the authors are not necessarily those of The National LOGO Exchange. Address all editorial correspondence to the editor at The National LOGO Exchange, Post Office Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor......................Tom Lough

Walls, Fences, and Camps

We see walls of every fashion built everywhere we look. Walls to keep things in, walls to keep things out. Physical walls, mental walls, social walls, emotional walls. And, we must add, professional walls.

When an educator, having become familiar with a certain aspect of the profession, becomes entrenched in a sort of "comfort zone," and refuses to consider alternatives or improvements, often walls are built to block change. Akin to the QWERTY phenomenon of Papert, the situation often leads to camps of stagnation and lethargy.

For a pointed example, consider the following scenario. A group of educators have discovered LOGO and are genuinely excited about its potential. Another group in the same school, having had experience with BASIC, forms a lobbying camp to persuade the administrators that LOGO is "not as good as BASIC."

Committees are formed to "study" the situation. Actions bog down. The communications between the camps take on a chilling hardness. The LOGO introduction never takes place.

What happens? The children are deprived of the most exciting educational opportunity now available. The educators are split by walls and fences into opposing camps. Both friendships and professional relations suffer.

This need not happen, if the situation is approached with a genuine desire to understand "the other side" and the underlying feelings which cause them to act as they do.

A tall order, true. But, the children ... the children ... ▷

Parent continued

CREATION

Another key to optimum learning is self-confidence. A student must believe in himself to function at peak capacity.

Again, I feel that LOGO surpasses the criterion. With the fresh approach of the child being the teacher and the computer waiting to learn, the child's self-confidence is raised because he is in control. He dreams up the project and instructs the computer what to do, where and how to do it.

So, as a child learns the basic principles through experience and how to apply them, he finds that he can make reasonably accurate predictions. This boosts his courage and confidence. The computer's speed offers immediate feedback regarding the accuracy of his predictions, making it easy to correct errors.

These two things make it possible to gain quickly much actual "experience" in a non-judgemental atmosphere, again boosting self-confidence.

As a valuable by-product, a child can develop the ability to express himself creatively, and to believe that his ideas count!

MOTIVATION

An additional key to optimum learning is a sense of purpose. This keeps the interest level high.

Again, LOGO excels. Today's child, receiving much visual stimulation, is receptive to this medium. But LOGO is more than entertainment. It captures the child's interest because it is his project and he is in control.

A certain need arises because of one of his goals, and he must either create a tool to use, or plan how to accomplish a certain task.

An excellent by-product is that the child learns to apply his knowledge to the world around him. He will be able to see the geometric shapes in even complex forms, and he will be able to estimate with reasonable accuracy the size of an angle and the relative length of a line. (Is not this a skill with which an artist captures a likeness?)

It seems to me as a parent, then, that LOGO offers major benefits relative to an optimum learning situation through exploration, creation,

continued on page 4

TIPPS for TEACHERS

by

Steve Tipps

Patterns and Repetition

The LOGO command REPEAT marks an important transition from control of the turtle on the screen to programming. Programming is the ability to conceive and follow actions which are not individually and immediately performed as are the FORWARD, BACK, RIGHT, and LEFT commands which allow the student to control the turtle's movements.

However, understanding REPEAT and using REPEAT demands recognition of patterns and how useful those patterns can be in describing the world.

BEGINNING PATTERNS

When objects or events are arranged in some repeating sequence, a pattern is formed. Children's understanding of the world according to Piaget moves from body to object to symbol. Patterns with which they work can also follow that developmental sequence. The children can stand in a line in some pattern such as:

BOY GIRL BOY GIRL BOY GIRL BOY ...

or, a little harder:

GIRL BOY GIRL BOY GIRL BOY GIRL ...

or, with chairs:

SIT STAND SIT STAND SIT STAND ...

or, in pattern by color of clothes:

BLUE RED GREEN BLUE RED GREEN ...

The variety of patterns which children can act out and the possible complexity of the patterns are almost infinite. Rhythmic patterns are another example at the body level:

CLAP CLAP (rest) CLAP CLAP (rest) ...

CLAP STOMP STOMP CLAP STOMP STOMP ...

or, with instruments being played:

BANG TING RASP BANG TING RASP ...

or, with animal sounds:

MOO WHINNY OINK WHINNY MOO WHINNY ...

The role of the teacher in pattern games is crucial. While the teacher may set up some of the patterns to alert the children to their existence, very quickly the children should be working in committees to invent their own patterns to challenge their classmates.

The examples chosen reflect a belief that active, body oriented learning is an important precursor to later symbol manipulation. Mary Barrata-Lorton's *Mathematics Their Way* (Addison Wesley, 1978) has an excellent chapter on patterns as does a booklet in the Nuffield Mathematics series. The examples which are found should only stimulate other exercises for the teachers and students to invent.

WHAT TO LEARN FROM BEGINNING PATTERNS

If you had to stop for a split-second and ponder any of the patterns above, you are alreay experiencing the learning which should be going on with patterns. Recognition that a pattern exists is an important skill. However, counter examples of random placement can also be used to point out that there is not always a pattern to things.

Exposure to patterns which vary in length, number of items, and complexity of items broadens the experience beyond simple AB AB patterns. Many commercial materials include only 2- or 3-item patterns or only present one "kernel," such as AB. Both practices simplify the pattern concept too much.

Children should be able to show where a pattern begins and ends (first and last elements), tell how many elements are in the pattern, describe how many repetitions are in the example, and add another repetition or fill in missing elements. Children should be able to create their own patterns and follow the sequence of other children's patterns.

The skills which are started with beginning patterns continue with object and symbol patterns.

OBJECT AND SYMBOL PATTERNS

Art materials are an obvious way to continue children's exploration of patterns. Circles, squares, triangles, and other shapes of many colors can be glued on strips of manila paper or sentence strips.

The obvious can be much more complicated than is first apparent. Patterns which focus on attributes

Tipps for Teachers continued

rather than labels and patterns which require analysis of more than one physical attribute will challenge students in the fourth or fifth grade.

For example, the pattern of the sequence RED SQUARE, BLUE CIRCLE, GREEN SQUARE, RED CIRCLE, BLUE SQUARE, GREEN CIRCLE requires analysis of two different subpatterns within a main pattern. Or, a pattern might focus on the number of sides and angles of irregular shapes rather than easy-to-recognize squares or octagons.

Multiple classification games with pattern blocks, beads, buttons, seeds, etc. are related thinking activities which encourage children to think abut two variables at the same time. Geoboards and pegboards can also be used for pattern creation.

When students are encouraged to look for patterns -- regularities in the world , they will become aware of many examples: architectural details on buildings, wallpaper and fabric, automobile design, shapes in paintings and sculpture, how supermarkets are arranged, letters in words (CVC), TV schedules, ad infinitum.

However, you may be wondering what all this has to do with LOGO. Recall that LOGO is another tool which helps the teacher create an environment for thinking.

Isolating LOGO from the rest of the goals of education and other curriculum areas would be foolish. Preparing children to understand the concepts imbedded in LOGO is essential. Patterns are an important thinking strategy which is related to the LOGO command REPEAT.

REPEAT, REPEAT, REPEAT

When children have full control of the turtle with the distance and direction commands, they will usually begin to set up a pattern of commands such as:

```
FORWARD 50
RIGHT 50
FORWARD 50
RIGHT 50
FORWARD 50
RIGHT 50
etc.
```

When the teacher observes that several children are using patterns to create figures and designs, the time has come to introduce the REPEAT command. The teacher might ask if there is a pattern to the series of commands. The teacher who has prepared students with patterns could suggest that a bracket be put around the kernel of the pattern, as:

```
[ FORWARD 50 RIGHT 50 ]
```

The question of how many repeats comes next. The child may say 12. Then the teacher could suggest

```
REPEAT 12 [ FORWARD 50 RIGHT 50 ]
```

Quickly, the student will recognize that the REPEAT statement saves a lot of effort and will get quicker results with the turtle.

Introducing LOGO commands in response to the needs of the children is very different from having a lesson on REPEAT. Glen Bull's Microworlds column in this issue gives greater meaning to this statement.

Good programmers constantly look for ways to take advantage of the ability of the computer to do the same action over and over and over very rapidly. Repetition is a prime programming concept which occurs in different ways. The REPEAT statement is only the first example of this powerful feature of LOGO. But using any of the repetitive ability of the computer is based on patterns. The next column will suggest some of the things which children can do with REPEAT before going further with LOGO. ▷

Steve Tipps is a professor at the University of Virginia, and conducts LOGO workshops for teachers throughout the eastern United States.

Parent continued

and motivation, provided that you, as teachers, catch the vision that computers are more than merely an expensive set of electronic flash cards. Parents are counting on you for this!

This is the unique challenge LOGO places before us. Explore! Create! Motivate! ▷

Sandy Towberman is an artist, writer, and musician who, with her husband, a Lutheran clergyman, is raising a daughter in Canton, Ohio.

In LOGO, as in life, the direction you are heading is often much more important than how far you go.

MICROWORLDS

by

Glen Bull

The Meaning of Meaning

Throughout formal academic training it is frequently necessary to master information which has a deferred application. In some instances the application may come relatively soon, as in the case of calculus in engineering school. In other cases the information may not be employed for years.

There is almost always a presumption that the subject learned will be of value at some time, whether immediate or deferred. Only a perverse instructor would claim to teach meaningless materials, and the phrase "meaningless learning" would appear to be almost a contradiction in terms.

Therefore, I was mildly surprised to find that in some studies children appeared to learn to read faster through a non-meaningful approach.

Two senses in which the word "meaningful" can be used may resolve the apparent contradiction of children learning to read more effectively by using a less meaningful approach:

(1) comprehensible, and
(2) of personal interest.

In the study on reading, the term probably refers to the first sense of the word. In this sense, meaningful vs. non-meaningful might be translated as delayed vs. immediate comprehension. However, differences resulting from delayed vs. immediate comprehension in reading acquisition might ultimately prove to be minor in comparison with the factor of the interest of the learner.

WRITING POWER

Both reading and writing are language skills which are acquired with great difficulty in comparison with the relatively painless acquisition of oral language. A typical child in the fifth grade completes an essay on "What I Did on My Summer Vacation" under duress. It's certainly nothing that would have occurred to him to do on his own.

continued

Written language ultimately becomes comprehensible to most children, but for many it never becomes meaningful in the second sense of the word. The problem is that written language simply isn't powerful enough at the time it is learned. Power in this context may be defined as allowing a person to do something they want to do without undue difficulty.

Written language may not be a powerful tool for a child if it does not allow him to accomplish a personal goal. With this perspective, it becomes clear that the struggle to teach a child to read may resemble a boy scout dragging a little old lady across the street. It is considerably easier to do if that is where she wants to go!

This is regrettable, but generations of teachers have long consigned themselves to the fact that much which is learned is only appreciated in retrospect.

It is apparent that few fifth graders will ever find an essay the most effective way to transmit information to their teacher. The paper is buried among thirty others, and the chances are that the teacher has seen similar efforts from classes in preceding years.

Further, the teacher doesn't write back, at least not in length. The most that can be expected is a cryptic word or two on the side. Under those circumstances, it's small wonder that most children don't find written language a very satisfying avenue of communication until later in life.

COMPUTER POWER

This raises the question of whether communication with a computer might be more satisfactory for a child. A computer, after all, isn't even human. However, it can respond.

The element lacking in written communications to teachers is that of turn-taking. There simply are not enough hours in the day for a teacher to respond to each child with a missive of equal length. Even if there were, it is doubtful whether a classroom instructor could find something fresh and interesting to say to each child.

In contrast, communication with an interactive computer system is turn-taking personified. The child commands the computer, and lets the computer know that it is its turn by pressing "return." The computer then responds to the best of its ability, and informs the child of its turn by returning a prompt.

continued on page 6

Microworlds continued

Further, the child is the one who controls the direction of the conversation. This means that a conversation with a computer need only occur if the topic is of interest to a child.

Is this type of activity meaningful? It certainly can be if meaningful is defined as "of personal interest." The programming skills learned in the process also will be of value to both the learner and society at a later date.

LOGO POWER

All of the above is true only if the child can use the computer to accomplish a personal goal. LOGO is particularly suited to use by both adults and children because it allows a great many activities to be accomplished with less fuss than other computer languages --- in short, because it is more powerful in many more applications.

This applies only if LOGO allows the child to accomplish a desired activity. It should be expected that some children will not want to draw squares, triangles, landscapes, or any of the other possibilities offered by LOGO's graphics capabilities.

It may be that some of these children will be engaged by the list-processing and the language-handling capabilities inherent in LOGO. However, there will be a residue of children who do not see the point of LOGO, and who will not want to do anything which LOGO allows them to accomplish.

For these children, LOGO is meaningless, at least in the sense of being of personal interest. Their avenues of interest will lie in other directions.

I believe the number of these children will be small, but a certain number is inevitable. The adult who forces a child to perform sterile exercises with LOGO in the face of this lack of meaning does not understand the meaning of LOGO. ▷

Glen Bull is a professor at the University of Virginia, and teaches LOGO courses at both the graduate and undergraduate level.

Mindstorms: Special Notice!

Seymour Papert's "Mindstorms" is now available in paperback from Basic Books, 10 East 53rd Street, New York, NY 10022, for $6.95.

Turtle Geometry continued

The first page of the text greets you with the announcement that turtle geometry is a mathematics designed for exploration. For those already acquainted with LOGO philosophies, this idea will be refreshingly familiar. This design is carried out to the fullest in the organization of the text.

Numerous examples and problems, both closed and open-ended (those are especially fun!), are sprinkled throughout. Moreover, they are grouped into levels of difficulty. "Regular" problems have no special designation. More difficult ones are marked with (D). The most difficult (impossible?) ones appear with (DD).

Even then, the authors displayed a sense of sympathy and support for their readers by adding a section of hints and another of more complete answers at the back of the book. This was done in a way designed to provide a gentle encouragement to those who took up the challenges.

The reader is cautioned not to read the book, but to do it! With so many opportunities for exploration provided, it seems unlikely that any reader could resist the invitation to participate. So, if I may offer a bit of advice, make sure you have access to a computer with turtle graphics (LOGO would be best) before you begin your trip through the incredible world of "Turtle Geometry."

The publication was designed to be a textbook at the advanced high school or college undergraduate level. But it is much more than that. The authors have incorporated ideas and concepts from many diverse sources and have interwoven them into a tapestry whose intricacy is not evident to a casual observer.

It would not be unreasonable to predict that this or similar texts will be used to teach the physics and mathematics courses of the future, with emphasis on exploration and doing, instead of passive learning. The dream of an information system containing laws of physics, theorems of mathematics, and rules of behavior which can be interrogated by students in their own way at their own speed comes one step closer to reality.

"Turtle Geometry," by Harold Abelson and Andrea diSessa, 1981, The MIT Press, Cambridge, MA.

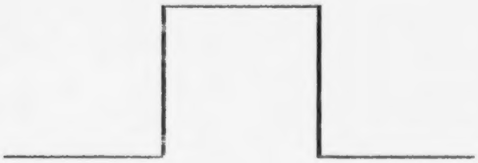
FORWARD 100! ▷

Fractal Fun with LOGO

Fractals are an interesting class of geometric objects. They can be studied and manipulated by first graders, and used by research physicists to explain extremely complex natural phenomena.

But the best thing about fractals is that LOGO can be used to draw them easily. Thus, the study and manipulation of fractals is within the reach of virtually everyone.

Let's take a look at a sample fractal to make sure we all know what we are talking about. Consider the shape below.

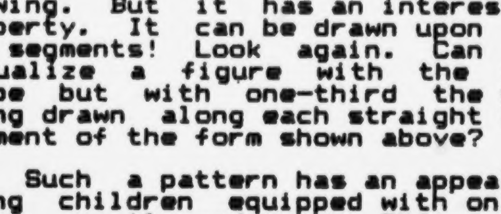


This is a rather simple line drawing. But it has an interesting property. It can be drawn upon its own segments! Look again. Can you visualize a figure with the same shape but with one-third the size being drawn along each straight line segment of the form shown above?

Such a pattern has an appeal to young children equipped with only a ruler, pencil, and paper. Refinements are easily constructed, until the first practical limits imposed by infinity are encountered.

Thus, fractals could be used as an extension of the ideas for learning about patterns presented in the Tipps for Teachers column in this issue.

Here is a drawing of the shape with one "repetition" of its pattern drawn instead of its original segments. Trace the outline carefully to find each of the five smaller figures superimposed upon the original.



continued

Can you visualize the form of the next "repetition" from the pattern above? If so, you have thus encountered the concept of a space filling curve. As the depth of the repetitions increases, the total length of the line of the figure increases, presumably without limit. And, the line fills more and more of a space with a particular shape.

It is an instructive exercise to attempt to write a graphics program in BASIC to draw this shape to an arbitrary depth of repetitions. In a very short time, the limitations of that language become evident.

However, it is possible to write a LOGO program of just a few lines to accomplish the task. All that is necessary is a good understanding of the power of LOGO's recursion.

The main idea to grasp is that a procedure which is called continues until it stops. Only then can the procedure which called it continue.

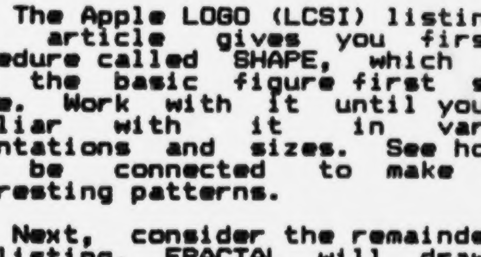
This concept is developed in some detail by Harold Abelson on pages 34-43 of his book, "Apple LOGO" (BYTE/McGraw-Hill, 1982). Abelson and diSessa also explore it in their text "Turtle Geometry," reviewed elsewhere in this issue.

The Apple LOGO (LCSI) listing in this article gives you first a procedure called SHAPE, which will draw the basic figure first shown above. Work with it until you are familiar with it in various orientations and sizes. See how it can be connected to make many interesting patterns.

Next, consider the remainder of the listing. FRACTAL will draw the SHAPE of a certain size to a desired depth of repetition. Note that the TEST for DEPTH = 0 will STOP a procedure if it is true. This is one of the keys to understanding how this procedure works.

Also note that, instead of FD :LENGTH along a side of the figure, FRACTAL calls itself with :LENGTH/3 and :DEPTH-1. Thus, its own shape can take the place of a side at a particular depth of repetition.

Here is what FRACTAL at a DEPTH of 3 looks like.

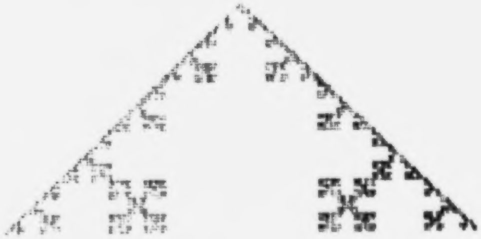


continued on page 8

Fractal continued

CENTER is just a housekeeping procedure to align the drawing horizontally in the center of the screen.

SURPRISE is the final result, all ready for you to use. Just as an opener, here is what the figure looks like at a DEPTH of 5. The surprise is the unexpected (?) appearance of a corner.



Now you can go back to some of the patterns you made with SHAPE and change SURPRISE to make some incredibly intricate designs.

```
TO SHAPE :LENGTH
   FD :LENGTH
   LT 90
   FD :LENGTH
   RT 90
   FD :LENGTH
   RT 90
   FD :LENGTH
   LT 90
   FD :LENGTH
   END
```

```
TO FRACTAL :LENGTH :DEPTH
   TEST :DEPTH = 0
   IFT [ FD :LENGTH STOP ]
   FRACTAL :LENGTH/3 :DEPTH-1
   LT 90
   FRACTAL :LENGTH/3 :DEPTH-1
   RT 90
   FRACTAL :LENGTH/3 :DEPTH-1
   RT 90
   FRACTAL :LENGTH/3 :DEPTH-1
   LT 90
   FRACTAL :LENGTH/3 :DEPTH-1
   END
```

```
TO CENTER :LENGTH
   CS PU HT RT 90
   BK :LENGTH/2 PD
   END
```

```
  TO SURPRISE :LENGTH :DEPTH
     TEST :LENGTH >243
     IFT  [ PRINT  [ TOO LONG. ]
STOP ]
     CENTER :LENGTH
     FRACTAL :LENGTH :DEPTH
     END
```

continued

Incidentally, if you have a desire to explore fractals in a more formal sense than that presented here, you are referred to "Fractals: Form, Chance, and Dimension" (W. H. Freeman pub., 1977), by Dr. Benoit Mandelbrot, the father of modern fractal study.

FORWARD 100! ...... ▷

From the NLX Mailbag

The article "Tipps for Teachers: Distance and Direction" written by Steve Tipps is excellent. Every paragraph includes a meaningful instructional strategy. I look forward to the next issue of The National LOGO Exchange.

Grace E. Shope
Abington School District
Abington, PA

TERC Announces Computer Workshops

Technical Education Research Centers Inc., a non-profit public service corporation well known for its LOGO involvement, has announced the dates and sites for its workshop series, Microcomputers in Education, for the fall and winter.

Special emphasis will be given to LOGO. However, other topics also will be covered, ranging from BASIC and PASCAL to Administrative Uses of Microcomputers. Machine language, graphics, and instructional use of computers round out the subject area.

These workshops are designed for professional development for educators at all levels, elementary school through college. Each workshop emphasizes hands-on experience with several different microcomputers.

Participants receive a hefty packet of practical reference materials. Special evening symposia are usually held to review current topics and issues in the field of computers in education.

The schedule is as follows:

October 19-21	Westport, CT
November 8-10	Cambridge, MA
December 7-9	St. Louis, MO
December 13-15	Boulder, CO
January 18-20	Tallahassee, FL
February 7-9	Washington, DC
February 17-19	New York, NY

For more information, write Ms. Sharon Woodruff, TERC, 8 Eliot St., Cambridge, MA 02138, or call (617) 547-3890. Mention that you read about them in The National LOGO Exchange. ▷

ISSN 0734-1717

The National LOGO Exchange

Volume 1 Number 3 FORWARD 100! November 1982

Molly Watt Writes for Teachers

The October issue of Creative Computing magazine contains an article well worth reading. "What Is LOGO?" by Molly Watt, appearing on page 112, is an excellent composition which presents the point of view of a teacher who has used LOGO in many different situations.

If you have been on the lookout for LOGO documentation to give to decision makers in your school systems, plan on reading Watt's article. It might be just what you have been needing.

She starts with an overview of LOGO and a bit of the historical development. Then she presents some examples of typical beginners work.

Next, she shares some of her personal experiences as a teacher in Amherst and Cambridge, Massachusetts. Her students had a wide age range, from 8 to 17 years.

Her case studies of the individual students reveal the true potential of LOGO, both of the language itself, and its accompanying teaching philosophy.

She then directs her attention to the development of the role of the LOGO teacher, and offers several specific techniques which were effective for her.

Lastly, she makes an extremely interesting comparison: what LOGO teachers say they teach versus what LOGO students say they learn. There are probably no big surprises in this comparison. But, the reader is likely to make a few new discoveries which may not have been apparent before.

In addition to the Watt article, Robert Lawler's "LOGO Ideas" on page 111 of the same issue contains a discussion of the passing of parameters which clarifies several points often misunderstood.

He also discusses the virtues of planning how to break a programming problem into parts, operating on each individually. ▷

Hawaiian Students Love LOGO Overlays

based on an idea submitted
by Elaine Blitman

Acetate transparencies can be used to create LOGO overlays which have many applications. Elaine Blitman, a supervisor for Punahou School in Honolulu, Hawaii, and her teachers and some 800 children are exploring the many possibilities offered by this idea.

To make a LOGO overlay, simply obtain a sheet of acetate such as that used for overhead projectors. Draw directly on it with a grease pencil or acetate marker, and place it on the computer monitor screen.

In most cases, the normal static cling will be sufficient to hold it in place. Otherwise, turning the monitor off then on once will usually help.

What sort of thing can be drawn on the overlay? That is left to the teacher (and the children). Mrs. Blitman suggests mazes, game boards, targets, and other similar shapes for starters.

Once the overlay is in place, the children can guide the turtle around, through, over, to, along, or behind the parts of the drawing, depending upon its nature and invitation.

A progression of overlay patterns suggests itself as part of the all-important development of turtle control. Perhaps the first overlay could be one with three or four numbered dots on it. The children are invited to move the turtle from HOME to dot #1, then to #2, and so on in succession.

Next could come an overlay with a simple closed curve in the center. The children are challenged to figure out how to make their turtle go around the outside of the closed curve one time, without necessarily trying to come close to the sides of the curve.

Perhaps next would be an overlay with a more complex numbered dots pattern, even one with a connect-the-dots picture which will develop as the turtle follows the correct path.

continued on page 2

The National LOGO Exchange, copyright 1982 Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly September through May, $25 per year, $45 for two years, mailed FIRST CLASS from Charlottesville, VA. Add $5 per year for addresses outside the United States. Single copy price $3. Opinions expressed by the authors are not necessarily those of The National LOGO Exchange. Address all editorial correspondence to the editor at The National LOGO Exchange, Post Office Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor......................Tom Lough

Hold Your Horses

A colleague asked me recently, "Do you know what's the hardest thing for me to do when I'm teaching LOGO?" I didn't receive a chance to respond before she continued, "I'll tell you what. It is to not intervene. Do you understand?"

I encouraged her to continue. She said, "When the children are working so hard with LOGO, and they have gotten bogged down with a problem, I have this overwhelming urge to rush right in with the solution. It is so hard to wait or to just ask a question instead."

The experience is common to many. We all have seen the excitement on the faces of children which comes from making discoveries or accomplishments all by themselves. And, we all at some time have seen slightly crestfallen looks when we offer the solution to a problem too soon.

Wait time is important. Children need time to think. In a situation as simple as directing a question to a child in class, it is difficult to wait through long agonizing moments for an answer. How often do we yield to the temptation of asking another (perhaps quicker) student before the first one has had a chance to think?

With LOGO, wait time is particularly important. We all need to work hard to identify that special moment between "I hate this stupid computer anyhow!" and "Don't tell me! I want to figure it out for myself!"

It is then, and perhaps only then, that the comment, the question, the hint, or even the answer will be most effective.

It is hard to hold your horses. But let's all try. ▷

LOGO Overlays continued

Next could come simple mazes and obstacle courses. By then, the children would have devised several of their own design, of course.

Perhaps the turtle could become a space rocket and visit various planets drawn on the overlay. Or, maybe the turtle is really an automobile navigating a street map on the screen.

Sprites could also be used in conjunction with the LOGO overlays, if you have the Texas Instruments version. The children will learn quickly that reasonable values for SETSPEED must be selected! However, if you TELL SPRITE 1 to CARRY :BALL and SETCOLOR :YELLOW, there appears on the screen a creature bearing a fairly close resemblance to PAC-MAN which many children can move about quite skillfully!

The following TI LOGO listing will provide children with a means of controlling a moving sprite:

```
TO E
   SETHEADING 0
   END

TO S
   SETHEADING 270
   END

TO D
   SETHEADING 90
   END

TO X
   SETHEADING 180
   END

TO CONTROL
   CALL READCHAR "K
   RUN SENTENCE :K [ ]
   CONTROL
   END
```

By entering CONTROL, children can then guide a sprite through a maze by pressing the E, S, D, and X keys at appropriate points. Note, however, that neither the sprite nor the turtle can "sense" what is drawn on the overlay.

For schools without a printer or a version of LOGO which will save the contents of the screen, the LOGO overlay idea could be used in reverse. Once a child has produced something on the screen, it can be saved and taken home to parents by placing a blank piece of acetate (or tracing paper, for that matter) on the monitor screen and copying the drawing directly with a marker.

Also, if the drawing is particularly challenging, it can be placed on other children's monitors with an invitation to try to copy it with their turtle!

continued on page 8

TIPPS for TEACHERS

by

Steve Tipps

Patterns and Repetition II

The previous column on patterns and repetition suggested many ways which teachers can encourage students to find and describe patterns in the environment. The REPEAT command serves as the way which LOGO allows the programmer to draw patterns and consolidate understanding of direction and distance. The amount of time spent with consolidating control of the screen turtle will be rewarded as children move toward procedures.

REPEAT allows the programmer to see the results of the same turtle movement done again and again. The simplest form of this is a single command inside the brackets:

```
REPEAT 3 [ FORWARD 20 ]

REPEAT 17 [ FORWARD 50 ]

REPEAT 100 [ BACK 6.7 ]
```

The additive nature of the turtle's microworld is clearly demonstrated by these simple commands. The turtle can also be made to pivot with a REPEAT command in a game called Spin the Turtle. The only command inside the brackets is a LEFT or RIGHT command:

```
REPEAT 20 [ RIGHT 11 ]

REPEAT 3 [ LEFT 15 ]

REPEAT 300 [ LEFT 20 ]
```

As children make the turtle spin they find that combinations of very large numbers only result in the same directions as smaller numbers. Very large numbers are superfluous because the direction of the turtle is limited.

Some children may stumble across the magic number of 360 which is the number of turtle turns in the Total Turtle Trip Theorem. (See, for example, page 24 of Turtle Geometry by Abelson and diSessa.)

Students can get involved in guessing games about the final orientation of the turtle as a result of REPEAT statements. The ability to use repeat with direction and distance separately will increase the possibility of using them together and with other commands.

STACKING COMMANDS and DEBUGGING

Distance, direction, and pen commands are still being used in the graphics (immediate) mode, so that feedback in the form of turtle movement is immediate with hitting <RETURN>. When children first learn that they can stack commands on a line:

```
    FD 20 RT 45 LT 33  PU BK 60 PC 3
PD FD 90 LT 45 PC 6 FD 70 RT 80 BK 66
FD 9 BK 91 RT 56 FD 81
```

they will often adopt this garbage approach to turtling. The initial fun of having the turtle do all these actions rapidly and in sequence can be counterproductive to using and recognizing patterns which do interesting things.

The teachers should encourage the child to look at single commands or at most two related commands on a line. This is very early debugging practice which encourages "mind sized" chunks of programming.

I would not say that a child should never stack up a line of commands. The idea that LOGO does whatever it is told in the order which is specified and that the sequence is unlimited for all practical purposes (save 255 characters on a line) are important LOGO understandings.

But children should recognize that it is usually very difficult to figure out which part of a very long line of commands was responsible for a particular part of a complex turtle drawing.

As children begin to draw things one command at a time in the graphics mode, they will begin to see that repeating the same combination of commands results in some particularly interesting designs. Closed figures and designs result if a pattern of distance and direction is created.

```
FD 38 RT 33         FD 100 LT 144
FD 38 RT 33    or   FD 100 LT 144
FD 38 RT 33         FD 100 LT 144
etc.
```

The patterns call upon the teacher to introduce REPEAT, enabling the child to explore the combinations of distance and direction commands at a faster rate.

```
REPEAT 22 [ FD 38 RT 33 ]

REPEAT 22 [ FD 100 LT 144 ]
```

The teacher must remember that s/he is not teaching formulas or giving out recipes for squares, stars or circles. Instead, s/he is only providing the format of the command which allows the students to create squares, stars and circles without

continued on page 4

typing each and every command separately. The recipe process is called "repetitive and redundant" by Glen Bull, while what you want to encourage is a process which is only "repetitive."

Collapsing many actions into a REPEAT command is the transitional step to understanding what programming is. Children will also try out putting garbage into the brackets. Again this may produce some interesting effects which the child can clean up.

My approach to this stage is to ask the child to describe or draw what is going to happen as the sequence is repeated. If they show understanding of the effect of the command and the sequence of action, I'm not concerned. They will be able to do other more coherent programming when they need to. If, on the other hand, the student seems to be mystified and is using REPEAT in some mindless way, I tend to guide them back to one or two commands in the brackets.

It would be nice to pretend that a teacher could only show REPEAT to those children who demonstrate a need for it. However, the classroom grapevine will pass along trade secrets. REPEAT gives so much power to the beginner that they will not be denied even if they are not in complete control. The teacher's job, as I see it, is to guide their thinking toward understanding the command.

ZIGZAG: AN ILLUSTRATIVE EXAMPLE

The next step is to control more commands within REPEAT. Zigzag patterns are particularly good because they have many variations of direction and distance — almost all of which result in pleasing designs.

I would suggest that zigzags be used to reinforce the ideas of body syntonicity and control which are at the foundation of LOGO philosophy. At a number of teacher workshops, when participants were invited to create a zigzag pattern, they began by creating a closed figure like those above.

Instead, they needed to walk through the pattern using conversation while playing turtle:

Walk forward,
turn to the right,
walk forward some more,
turn to the left.

Several of them tried to go directly to REPEAT statements rather than working out a design command by command.

```
FORWARD 50                  FORWARD 40
RIGHT 39          or        RIGHT 120
FORWARD 77                  FORWARD 40
LEFT 22                     LEFT 120
```

Working ideas out in the immediate mode is very important and is a prime advantage of LOGO. Only after a design is achieved should the REPEAT statement be used.

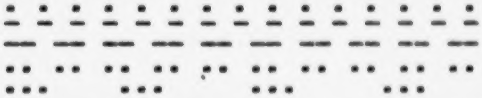
```
REPEAT 900 [ FORWARD 26 RIGHT 98
FORWARD 40 LEFT 100 ]
```

All manner of zigzags, wiggles and steps are created with the REPEAT format. The initial direction of the turtle can transform the appearance of the pattern. Overlaps and cross-hatching are often quite striking. As zigzags are created, programmers should become very familiar with the three elements of the REPEAT command and the effects of changing the repeater or the pattern.

PROJECTS WITH REPEAT

If the teacher has prepared students with pattern finding and making, and if the students are very comfortable with REPEAT statements, they should be ready to use REPEAT to create many of their own patterns.

I like to have the class create a set of patterns on tag board or sentence strips. Among the most frequent are dotted and dashed lines using PENUP, PENDOWN, and different distances in a REPEAT command.



Later, the dots and dashes could turn into Morse code or an idiosyncratic personal code which children can program.

Other designs might incorporate turtle moves such as resetting the turtle at the side of the screen, flipping it upside down, and spinning it to create a star effect or rays from a center position. (See "LOGO on Wheels" elsewhere in this issue, for example.)

Each child should keep a record of different designs and statements. Challenge designs can be posted as puzzles for other children to solve. Of course, not all the solutions will be exactly alike, but this will help the children compare their different approaches.

So far, positioning has been included in the REPEAT. However, recognition that the position and the pattern serve two different purposes

continued on page 8

MICROWORLDS

by

Glen Bull

Is LOGO a tool?

LOGO was intended to be a language with a low threshold and a high ceiling. It has been used effectively in classes ranging from kindergarten to physics courses at the university level. This is a wide range, but use of a computer language as a tool represents an even higher threshold of use.

LOGO has been used by many children to satisfy personally meaningful goals. In that sense, it is a meta-tool. It might be said to be a tool for learning how to use tools. However, this begs the question. A more meaningful question might be whether LOGO can be used as a tool in the context of adult goals in the work-a-day world, in the sense that COBOL is commonly used to develop business applications.

SOME CONSIDERATIONS

The physically and communicatively handicapped constitute a population where dramatic gains in quality of life could be achieved. The use of technology to gain control of the environment and establish communicaiton with the world is potentially one of its most humane uses.

The results of research in this area have not filtered down to the bulk of the population it will ultimately help. For example, far less than ten percent of the cerebral palsied population with communicative handicaps has access to technology potentially available to them. The limits on the real utility of a prosthetic device are dependent upon accessibility. If access to, say, eye glasses were limited by a price of $30,000 a pair, many of us would lead far less productive lives!

One key to driving down the cost of devices such as augmentative communication systems is utilization of general-purpose hardware. The economics of large-scale production runs is well known. As a matter of simple fact, a device developed for a specialized population may require an end cost ten times higher than a similar device developed for the general population.

A great deal of off-the-shelf hardware for interacting with and controlling the environment has been developed in the past two years. Devices which plug into the bus of the Apple computer, for example, control electric appliances and light switches. A reasonably intelligible speech synthesizer with an unlimited vocabulary is available for about $300. In contrast, the cost of an equivalent synthesizer in 1980 might have been $3000.

LOGO LIMITATIONS

LOGO is a language which should be rich with possibilities for interaction with the outside world. In the LOGO working papers (Bibliography of LOGO Memos, MIT LOGO Group, 545 Technology Square, Cambridge, MA 02139), there are references to dancing floor turtles, control of music boxes, use of robot arms to control yo-yo's, and other external devices. In this conception of LOGO, there should be ample capabilities for controlling general-purpose hardware which might be used by special populations.

This concept has not always survived translation into implementations on inexpensive microcomputers. In some cases, it has been necessary to sacrifice communication between LOGO and the outside world, except through the keyboard. In these instances, speech synthesizers, joysticks, and other external devices are available, but there is no way to tell LOGO about their existence.

In other implementations, it has been possible to include provisions for access to input and output (I/O) ports to the outside world. However, some of these implementations make no provision for calls to assembly language subroutines. This limitation makes it difficult for the user to develop specialized I/O drivers. To put it another way, lack of this capability places a very definite ceiling on the use of LOGO to interact with the outside world.

All current LOGO implementations on microcomputers (to my knowledge) have another constricting factor that limits use of LOGO as an adult tool; they are somewhat cramped for memory space. LOGO is more sophisticated than many languages, and hence occupies considerable memory. In fact, it was difficult to squeeze LOGO into the address space of an eight-bit microcomputer at all.

However, this necessary limitation on the size of programs which can reside in memory at a given time does hamper some of the more ambitious uses of LOGO as a tool for the handicapped.

continued on page 6

Microworlds continued

Much of the original work with LOGO was done on Digital Equipment Corporation minicomputers. It might be possible to evade some of the limitations I have mentioned by acquiring this version of LOGO. However, use of a minicomputer would defeat the purpose of increasing access to technology for the handicapped, since minicomputers carry hefty price tags.

At this stage, LOGO in some implementations on microcomputers is tantalizingly close to being an effective tool for real-world applications, but falls just short. The sensible thing would appear to be to develop these types of applications in some other language, such as Forth or Pascal.

LOGO UTILIZATION

However, there is another aspect which must be considered. The needs of the disabled vary considerably, and, in the best of all possible worlds, prosthetic devices would be customized. In fact, many such devices are customized of necessity since it is the only means of making them remotely usable. Customization in the case of a microcomputer implies an individualized program for the patient.

Currently, there are not enough engineers and programmers to go around, and there may not be for the forseeable future. This drives the cost of specialized programs up, if it is possible to obtain one for a handicapped individual at all. The cost may be less than that for equivalent individualized modifications in hardware, but the cost is still unaffordable in many applications.

It is not possible to teach clinicians to program interactive devices in Pascal. Many of them are capable of learning how, but there is not room for inclusion of this information in the curriculum. Computer scientists require several semesters of work to reach this stage. By the time a clinician took the required number of courses, it might be more sensible to go on and get a degree in computer science! Of course, some clinicians do add these classes to their regular courseload, but this is not realistic for most students.

However, LOGO is powerful enough and has a structure which permits a clinician to develop augmentative devices for the disabled in less than a semester of work. Memory limitations hinder expansion of these devices, but judicious use of the disk as virtual memory makes it clear what the potential could be.

Even when the programming concepts require prevent the clinician from completely developing a project in finished form, the process of working through some aspects provides insight into the task of the engineer. This, in turn, leads to improved communication between the clinician and the engineering team.

Currently, there are a number of 16-bit microcomputers which have expanded memory capabilities. It also would probably be easier to implement LOGO on these microcomputers, since development of a program becomes increasingly difficult as the capacities of a machine are stretched to the limit. The increased power of chips such as the Motorola 6800 microprocessor should decrease the difficulty of adapting LOGO to these systems.

On the other hand, the sophistication of LOGO makes it a complex language to develop, and hence more difficult to implement than a simpler language such as BASIC. This, in turn, decreases the probability that it will be developed on the next generation of microcomputers.

If it is developed for one of the 16-bit generation of machines, the development team may not understand the importance of making it possible for LOGO to interact with the outside world. Even if they do, technologic quirks may make it prohibitively expensive to develop some of the more desirable features of LOGO on the system adopted.

BACK TO TOPLEVEL

Thus, in answer to the question "Is LOGO a tool?" the reply must be "It almost is." or "It could be." LOGO in its abstract form as a programming philosophy foreshadows an eminently usable tool. In the practical requirements of the real world, the capabilities of 8-bit microcomputers on which LOGO has been implemented to date fall just short of providing the power required.

The limitations of memory space and insufficient processor speed are hardware restrictions rather than conceptual difficulties. Recent history suggests that hardware restrictions can be erased by technologic advances. From that point, it is a matter of creating a version of LOGO which makes use of this newfound capability. ▷

Glen Bull is a professor at the University of Virginia, and teaches LOGO courses at both the graduate and undergraduate level.

LOGO on Wheels

The power of LOGO to help in the investigation of phenomena present in the physical world was made evident to me recently.

As I was driving home one night, I saw ahead of me a strangely-behaving light crossing the road. It bobbed and weaved up and down in a herky-jerky fashion which commanded my attention.

It turned out to be a reflector mounted on the spokes of a bicycle wheel. (Whew!) Aside from making sure that I saw him, the bike rider also gave me something to think about.

What if a child observed the same motion and wanted to investigate it further with LOGO? How might the process take place?

By observing a bicycle wheel in slow motion, a child might note that the center of the wheel moves smoothly in the direction of travel while the spokes, reflector, rim, and tire rotate around it. The first challenge might be to make the turtle move like the axle.

By playing turtle, the child discovers that, by moving a little toward an object, and then turning some, then moving toward the object again (even if not facing the object), and so on, something close to the desired motion is produced. A spinning dancer crossing a room has a similar motion.

```
TO MOVE :STEP :TURN
     PENUP
     RIGHT :TURN
     SETX XCOR + :STEP
     MOVE :STEP :TURN
     END
```

The above MIT LOGO (Terrapin/Krell) listing might be one way to show this. MOVE 5 5 will make the turtle behave as if it were glued to the axle of an invisible wheel which is rolling across the screen. The turtle spins (rotates) and moves linearly (translates) in a uniform manner.

However, the bicycle reflector was not on the axle, but near the rim. How could this be represented with LOGO?

The turtle might mark the <u>location</u> of the reflector at successive moments. The turtle on the axle could be imagined to be facing the reflector at all times. Thus, it could dart forward, mark where the reflector is, and return to the axle in time to turn through the angle.

```
TO DOT
     PENDOWN
     FORWARD 1
     BACK 1
     PENUP
     END
```

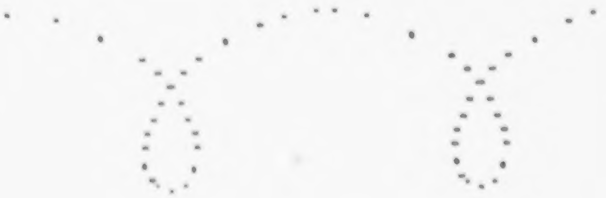
The above listing will make the turtle draw a dot at its location. Now to have the dot drawn at the location of the reflector.

```
TO REFLECTOR :DISTANCE
     PENUP
     FORWARD :DISTANCE
     DOT
     BACK :DISTANCE
     END
```

The REFLECTOR procedure needs to know how far the reflector is from the axle. This is designated as the variable :DISTANCE. Now let's integrate what we have done into the MOVE procedure.

```
TO MOVE :STEP :TURN :DISTANCE
     REFLECTOR :DISTANCE
     RIGHT :TURN
     SETX XCOR + :STEP
     MOVE :STEP :TURN :DISTANCE
     END
```

MOVE 5 10 40 will draw a picture similar to this:



This is just about what the path of the bike reflector looked like. The child has taken a reasonably complicated physical situation and has represented it well and quickly with LOGO in familiar terms.

Additionally, there has been ample opportunity for exploration along the way. Extremely interesting pictures and patterns can be generated by varying the parameters. It is also fun to choose values for the parameters, sketch out the predicted result, and then run the procedure. There are usually a few surprises!

But the story does not have to end here. The path drawn above is <u>not</u> exactly correct. Assuming the bike wheel rolls without slipping, there is a relationship between the angle and the horizontal distance the wheel travels in each cycle of MOVE.

continued on page 8

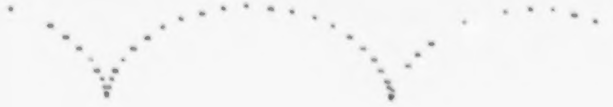
Wheels continued

In particular, the incremental horizontal distance (called :STEP here) is equal to the product of the wheel radius times the angle of the incremental turn (which must be expressed in radians instead of degrees).

For simplicity, let's assume the reflector is located right on the rim of the wheel (that is, :DISTANCE is the wheel's radius). Thus, if MOVE is rewritten accordingly, it would look like:

```
     TO MOVE1 :STEP :DISTANCE
          REFLECTOR :DISTANCE
          MAKE "TURN  (:STEP * 180)/
(:DISTANCE * 3.14159)
          RIGHT :TURN
          SETX XCOR + :STEP
          MOVE1 :STEP :DISTANCE
          END
```

Now we have a more accurate representation of the actual physical situation. MOVE1 5 20 will draw a rather nice looking curve similar to the one below.



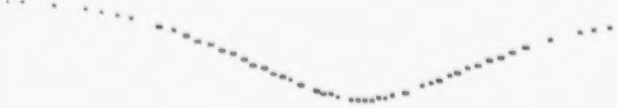
This is the path of a point on a rolling circle, one of a family of curves called CYCLOIDS which can be generated with MOVE1.

A further refinement might be in order. The reflector is usually mounted on the spokes of the wheel, not on the rim. Thus, the :DISTANCE from the axle to the reflector is less than the radius of the wheel.

By using the variable :RADIUS, we can write MOVE3 (my MIT LOGO would not recognize MOVE2 for some unknown reason):

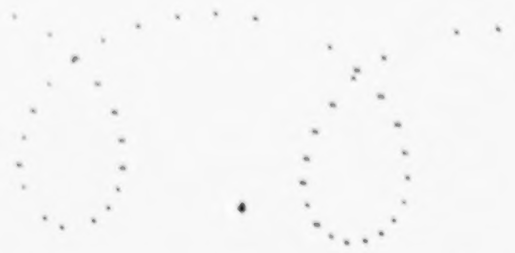
```
    TO MOVE3 :STEP :DISTANCE :RADIUS
         REFLECTOR :DISTANCE
         MAKE "TURN  (:STEP  *  180)/
(:RADIUS * 3.14159)
         RIGHT :TURN
         SETX XCOR + :STEP
         MOVE3 :STEP :DISTANCE :RADIUS
         END
```

By keeping the :DISTANCE less than the :RADIUS, we can get smoother patterns such as MOVE3 5 20 50:



continued

But, now we can investigate an "un-physical" situation in which the reflector is mounted on the wheel at a :DISTANCE greater than the :RADIUS! MOVE3 5 50 20 produces a picture thus:



Many everyday situations such as this are readily adaptable to investigation with LOGO. Look around. You might surprise yourself. Or, better yet, challenge your children to look around! And get ready!

FORWARD 100! ...... ▷

Overlays continued

The potential of this idea is limited only by the imagination of the teacher and the children. Let us know of any extensions or special applications you invent or use which you would like to share with others.

A special thanks to Mrs. Blitman for sharing this idea. She credits Theresa Overall of the Lamplighter School in Dallas, TX, as the original source of the LOGO overlay. ▷

Tipps continued

will be important as children move toward procedures and more modular programming.

Children who are skilled with REPEAT conceptually and functionally should be ready for the great leap into creating procedures. But, using REPEAT in the immediate mode is a programming and debugging skill which should not be discarded. ▷

Steve Tipps is a professor at the University of Virginia, and conducts LOGO workshops for teachers throughout the eastern United States.

With LOGO, the child becomes the teacher, and the teacher, the child.

ISSN 0734-1717

The National LOGO Exchange

Volume 1 Number 4 FORWARD 100! December 1982

Abelson's LOGO Books Are Winners!

If you have been looking for a useful LOGO book to help you get a handle on what LOGO can do for your Apple and you, be sure to get a copy of Harold Abelson's Apple LOGO (for use with the LCSI version) or LOGO for the Apple (for use with the Terrapin and Krell versions).

It is more than just a reference book. While Abelson teaches you something about LOGO, he also shows you what you can do with it.

Assisted in editing by Bruce Roberts and Dan Watt, Abelson has produced a well-organized, clearly written work which explains LOGO's "simple but complex" workings in an easily understandable manner.

The book starts with a look at the basic turtle commands and error messages. Then, it moves on to procedures and modes. Variables, repetition, and recursion are treated next, all with excellent examples to try. His detailed explanation of the recursion process brings light to what many regard as an extremely confusing matter.

Much attention has been given to turtle geometry elsewhere. Abelson chose to devote the majority of his book to non-turtle LOGO work. If you have heard that LOGO is "just turtle graphics," perhaps you should spend some time within these pages.

Numbers, words, and lists are given lots of attention. Example procedures and their development are presented for consideration. Sample quizzes, random sentence generators, and the evolution of a procedure to play and win the game of NIM make objects worthy of study.

The most fascinating part to me is in the advanced use of lists, particularly in the use of procedures as data. The mere fact that LOGO can write and rewrite programs and then run them is almost beyond appreciation. Abelson brings the reader face to face with this boggling concept in such a way that one cannot help but understand.

An easy-to-use glossary and a thorough detailed comparison between the Texas Instruments LOGO and the LCSI or MIT (Terrapin/Krell) Apple version round out the book.

continued on page 8

Brookline Students Hunt LOGO Bugs

by

Jay Sugarman

When I first heard the word LOGO, I thought it must have something to do with the emblem on the Izod clothing line! I couldn't understand how little alligators related to the school curriculum. Maybe if we were located in the Everglades, but certainly not in Brookline, Massachusetts.

Well, much to my delight, I found out that LOGO was the name of a computer language. This was three years ago. What I would like to share with you is how I now see LOGO fitting into the overall curriculum, and how I have come to relate this to my fourth-grade students, as a result of my experience.

To me, the most beneficial and exciting feature about LOGO is its emphasis on the problem solving process. LOGO not only encourages but demands that students become active participants in the learning process. They come to view the computer as a tool, as something they can control and adapt for their own purposes.

By stressing the educational value of the "debugging process," LOGO teachers can help students not only come to appreciate and understand the process of problem solving, but also to develop a comfortable presence around the computer. The students are not intimidated by the computer (as are some adults!), and are able and willing to devote their time and energy to writing and correcting their programs.

HOW TO SOLVE IT

To help my students to conceptualize the problem solving process, I begin the school year by talking with them about the four stages of problem solving (see George Polya, How to Solve It, 1945):

1. Understand the problem.
2. Plan what to do.
3. Carry out your plan.
4. Examine the solution you obtained.

continued on page 2

The National LOGO Exchange, copyright 1982 Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly September through May, $25 per year, $45 for two years, mailed FIRST CLASS from Charlottesville, VA. Add $5 per year for addresses outside the United States. Single copy price · $3. Opinions expressed by the authors are not necessarily those of The National LOGO Exchange. Address all editorial correspondence to the editor at The National LOGO Exchange, Post Office Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor......................Tom Lough

Christmas and ABC's

One of the things I always have wanted to do was to teach my turtle how to draw all the letters of the alphabet on the screen, in any size, at any position, in any orientation.

I had visions of providing children with the means of quickly creating interesting patterns and displays of letters and words. News bulletins and short compostions would appear easily. Suddenly, spelling might become more important. Writing might take on a whole new meaning. Creativity in composition could open into meadows unimagined.

A note in a prominent publication indicated that the memory of the Apple II would not provide room for the LOGO procedures required to draw all 26 letters of the alphabet. Initially discouraged, I decided to try to do it anyway.

The task is now complete! It is done. And what fun it is to create letter and word patterns on the screen! (Incidentally, I have been told that other people have also succeeded in this task.)

Imagine your name spelled upside down or in a wheel or other shape! Think of all the spelling activities you could help your children invent!

Here is our Christmas gift to you. If you would send us a self-addressed stamped envelope, we will send you a listing of the procedures to draw all 26 ABC's. Be sure to specify whether you have the LCSI or the MIT (Terrapin/Krell) version of LOGO.

Mail your request to NLX ABC's, Box 5341, Charlottesville, VA 22905.

Merry Christmas from each of us at the NLX! May 1983 be the "Year of the Turtle" for all of you!

FORWARD 100!!

▷

LOGO Bugs continued

After discussing this general framework, I then point out its relevance in each of the major disciplines. In reading, I talk about decoding the author's writing and analyzing the content. For writing, I focus on the writing process: the continual revising and editing that occurs from one draft to another.

In mathematics, I rely on word problems and real life situations and how to apply arithmetic processes. For LOGO, I refer back to the revising and editing process involved when writing and explain how these features also extend to the writing of computer programs.

LOGO BUG HUNT

Specifically, one of the most successful ways I've related this "debugging" aspect, the problem solving process, is by creating programs with deliberate errors and then having the students try to find and correct them. The children enjoy these activities and eagerly create their own programs with "errors" to see if their peers can find and correct them too.

Another activity I've used to convey the debugging approach is to design a maze on graph paper which can be solved by writing simple LOGO commands such as FORWARD 30, RIGHT 90 and the like. I then provide the students with the maze and a solution procedure which contains incorrect directions. It is up to them to reread the program, find the errors, and then write the correct solution for traveling through the maze. Their solution is easily tested by use of a LOGO overlay (see Hawaiian Students Love LOGO Overlays, NLX, November 1982).

While not a specific activity, I stress to my students the value of orally explaining both to themselves and to a friend what it is they are thinking when they are debugging a program. From my experience, this technique helps them become more involved and reflective participants in the problem solving process.

While I no longer associate alligators with the word LOGO, I'm continually reflecting about how it relates to the overall school curriculum. I am pleased with the progress both my students and I have made during the past three years. We just have to keep working at getting the "bugs" out!

▷

Jay Sugarman, Ed. D. Curriculum and Instruction, is a teacher in the Brookline (Massachusetts) Public School System, and participated in the Brookline LOGO Project.

TIPPS for TEACHERS

by

Steve Tipps

Getting Started

All of the "good" educational ideas in the world are worthless unless they are tested in real situations -- which is to say with real teachers, real children, and real classrooms. LOGO is in the early stages of making the transition from an academic idea to the real world limitations of time, money, and energy.

Getting started with LOGO and some of the many considerations in adoption of LOGO as a central part of a computer education program is the topic of this column. The questions and the tentative answers are drawn from the work that Glen Bull and I have done with fifteen fourth grade teachers in the Albemarle County Virginia Schools.

The answers are only tentative because one of the outcomes of the LOGO instructional development project will be a classroom guide. The teachers are engaged in finding better answers to the questions as the year goes along.

WHY LOGO?

Readers of the NLX probably already have their own answers to this question. LOGOphiles will cite as reasons for LOGO:

1. Having the child in control of the computer.
2. Using the computer as a tool to create interesting results in the form of graphics and language.
3. Exploring mathematical and geometric concepts in an exciting, technological environment.
4. Emphasis on problem solving and critical thinking in a context which leads to good programming principles and habits.

In the first and second cases, LOGO is an end in itself. LOGO is also a means to the ends in the third and fourth.

In the continuing debate regarding "computer literacy," LOGO is unique because it provides actual computer competency. Compare LOGO with approaches which focus upon isolated information about hardware and history of computing.

Such items are of little good or merit if the computer isn't being used to create interesting and useful projects. They are just the kind of nonsense being promoted in the name of computer awareness which LOGO should make passe. When children know how to use the computer, understanding some of the whys and hows, RAMs and ROMs, technology and terminology make more sense. But ... LOGO also demands a different kind of teaching.

HOW MUCH DO TEACHERS HAVE TO KNOW IN ORDER TO TEACH LOGO?

The answer is a good deal more than it takes to plug a child into a computer to drill math facts or answer capitals of states. But the goals are also much different and the professional educator will see the computer as a new tool for achieving the higher aims of education -- critical thinking and creativity -- rather than a new delivery system for the lowest objectives -- rote memorization and recitation.

Most needed for teaching LOGO are teachers who are willing to learn. The simple commands which allow almost immediate access to control over the turtle are somewhat deceptive.

The commands comouflage certain very sophisticated programming concepts -- control, repetition, levels of interest, variability, generality, modularity, extensibility. These concepts are not to be defined but experienced as the teacher becomes a learner again.

The same mistakes and struggles which the teacher goes through will make them more understanding and responsive to the frustrations of learning. Remembering when we didn't know how to read or multiply or remembering how we learned is difficult. Working with a programming language, however, is a constant learning experience which starts with the first command.

After the willingness and will to learn are accepted, several different ways to build skills are possible. One way is to learn with the children by keeping a LOGO manual such as Abelson's at your elbow. (See review elsewhere in this issue.)

Trial and error are certainly within the philosophy of LOGO and many teachers and parents and children are going about learning LOGO just this way. However, some of the long-range benefits of learning LOGO may also be slighted in this approach.

When Dr. Carlos Guttierez, Superintendent of Albemarle County Schools, decided with his staff that LOGO would be the basis of the elementary computer education pro-

continued on page 4

gram, he recognized that he could not simply decree, "LOGO!" and leave it. Staff development was an important part of the plan.

Glen Bull and I agreed to work with the teachers in the project to guide them in understanding the programming concepts embedded in the language and in making instructional decisions about implementation.

We already have had two all-day workshops and seven weekly two-hour meetings with the teachers. In that time, the teachers have developed conceptual and language understanding of LOGO by creating many turtle graphics. They also have learned something about terminology and technology in ways that directly affect their ability to create, save, and print their work.

After about twenty hours of instruction over two months, we felt that they had the necessary skills to begin work with children and to begin creating microworlds for themselves. (Teachers themselves are much more confident about the former than the latter.) The concerns and questions which teachers expressed about implementation and the guidelines proposed may be useful in considering your own plans.

HOW MUCH TIME DOES IT TAKE TO TEACH LOGO?

The most limited resource in a school day is time. If LOGO is considered to be a new subject, then everything else must be squeezed. Instead, teachers have been encouraged to think of ways to squeeze LOGO into the day by not having computer lessons per se.

For example, the ideas of distance and direction are just as appropriate in map study and mathematics. Patterning can be studied in science and language arts. Building the concepts should not be done in isolation from the regular curriculum.

However, three kinds of time need to be planned for: instructional time, independent time, and consulting time.

In instructional time, the teacher would guide the concept or command which is needed by most of the children to get started or progress with LOGO. Instructional time should be very limited. Group times such as those typically found first thing in the morning, before or after lunch, and just prior to departure should be capitalized on rather than a set LOGO lesson time. This time could also be used for sharing of projects.

The focus of the project is independent time in which the children will work on the computer in pairs (or triplets) to develop understanding of skills and invention of projects or applications. The teachers have analyzed the daily schedule so that times when students are doing seatwork, learning centers, or other independent work, they also can be scheduled on the computer.

At first, periods may be rather short (10-15 minutes), but later, 30 minute sessions (some solo) would be needed. Our teachers worked out two day cycles around reading and math groups. 80% utilization of the computer throughout the day was the goal.

Consulting time, or over - the - shoulder time, is the most difficult time to arrange because of the demands placed on the teacher. The luxury of looking over the childrens' shoulders and answering questions and challenging thinking may be the most important teaching time in any subject. Thomas Gordon of TET calls it "optimum time." Some teachers have worked out schedules for consulting time early in the morning. How this will be worked out in the long run is not certain.

HOW SOON DO I INTRODUCE NEW COMMANDS? HOW DO I KNOW IF THE CHILDREN ARE REALLY LEARNING ANYTHING?

Our tentative schedule for the year was to introduce turtle and pen commands (concept of control) and stay at that level for the first month to six weeks. Then teachers would introduce REPEAT (concepts of repetition and indirect execution) as the children need it. Procedures (concepts of delayed execution, programming, naming, and invoking) would follow and depend on needs.

Finally, master procedures (concepts of embedding) and modularity) would be introduced. We had not set this as an absolute limit on fourth graders' ability with LOGO. Instead, it was adopted as a tentative goal which would provide enough practice and experience so that more elaborate problem solving and programming in LOGO would have a firm base.

Fourth graders definitely can type more elaborate recipes into the computer using LOGO than are implied in this sentence. However, our rule is that if children cannot read and describe the actions of their program and debug it when there are problems, they are not conceptually sophisticated enough to understand the recipe.

continued on page 8

MICROWORLDS

by

Glen Bull

Languages for Implementing Solutions
and
Languages for Finding Solutions

A programmer from another planet might be mildly surprised to find that Earth's best-selling program of all time, Visi-Calc (a so-called electronic spreadsheet), was conceived by a novice. It might be said that this was simply beginner's luck. However, I don't think so.

LANGUAGES FOR IMPLEMENTING SOLUTIONS

The characteristics of a tool are most apt to be shaped by the needs of the individual creating it. A computer scientist needs languages with characteristics which enforce programming rigor. In a professional environment the ultimate goal is error-free code. The administrators of a bank are seldom intrigued by the improbable combination of circumstances that caused an account to be off by exactly four decimal places. Neither is the account holder, if the shift is to the left!

Thinking first through a problem tends to decrease the chances of unforeseen errors. This general method of using computers might loosely be said to be a top-down structured approach to programming. Making things up as you go along is frowned upon in circles in which payment is tendered for the end product.

A language characteristic which flows from the need to preplan a program is the variable declaration. A variable declaration is a list of variables which will be used in a program, and is similar to the list of ingredients at the beginning of a recipe in a cookbook.

The origins of the variable declaration are partly due to technologic considerations, as is the case with aspects of most software. A variable declaration also requires the user to think about the program before writing it. In this programming environment the language is a tool which is used to implement a solution which has been planned before the keyboard is touched.

Many aspects of languages designed for introduction of programming to aspiring computer scientists discourage such a trial-and-error approach. These languages, Pascal for example, are compiled. A compiled language makes it difficult to go through a development cycle rapidly since the editor, compiler, and loader may all be separate programs which result in a time delay when they are accessed.

These characteristics improve the efficiency of the language, and hence are due in part to engineering constraints. They also coincidentally penalize the numerous small changes associated with trial-and-error program development. Even if the technologic efficiency associated with compiled languages were not an issue, these features might still be viewed as positive rather than negative to the extent that they encourage good programming practice.

ELECTRONIC SCRATCH PADS

The electronic worksheet provides a different perspective. Visi-Calc is an electronic spreadsheet. It allows numbers to be entered into rows and columns. For example, a household budget consisting of five categories tabulated by month could be created.

	Jan	Feb	Mar	Apr	AVG
Food	175	225	240	180	205
Clothing	60	20	130	50	65
Rent	250	250	250	250	250
Utilities	90	80	75	63	77
Car	30	23	48	19	30
TOTAL	605	598	743	562	627

After the figures are entered, Visi-Calc allows each column to be totaled. Note that the figure at the end of each row is not a simple summation, however. The total amount spent on shelter for the four months was $1000, but the summary at the end of the row is the average ($250) rather than the total amount spent. In this case, the formula of the total amount divided by the number of months has been used to automatically generate the figures in the right-hand column.

Other ways of looking at the figures could be accomplished automatically. For instance, rent and utilities could be combined into a single category labeled housing.

Housing	340	330	325	313	327

The point to be observed is that experimentation is possible without undue penalty. A built-in inflation factor of 10% for utilities could easily be built into a month-by-month projection for the following year simply by multiplying the entire row by 1.1, for example.

continued on page 6

Microworlds continued

No repetitive calculations are performed by hand. Rather, a formula describing the calculations desired is written and Visi-Calc performs the operations outlined automatically. This ease increases the probability that many changes will be made.

Of course, some forethought invested in the budget categories chosen will be repaid, but an idea sparked by a preliminary set of results can be incorporated easily into the model. The presence of a resident editor makes alteration of the figures themselves easy as well.

A penalty of facilities which enhance ease of use is that resources such as memory are not used efficiently. Many users of Visi-Calc find that a complete application cannot be fully developed on a microcomputer system with 48 kilobytes of memory. Therefore, a spreadsheet may have to be broken up into several pieces handled separately.

A Pascal program could be written which would accomplish the same task with far less memory, and hence it could in a sense be considered more efficient in the use of computing resources - though not necessarily more efficient in conservation of human resources as time. The next generation of microcomputers now being sold with a quarter to half a megabyte of memory may alleviate some aspects of the efficiency problem.

LANGUAGES FOR FINDING SOLUTIONS

Visi-Calc is the equivalent of electronic scratch paper. It allows the user to doodle and muse over the merits of one approach versus another. The very name "Visi-Calc" is a truncation of "visible calculator." There are environments in which experimentation is appropriate. In these environments the computer is used as a tool for finding the solution, as well as for implementing it.

The non-obvious characteristic which Visi-Calc and LOGO have in common is that they both encourage experimentation without penalizing it. This is the advantage of scratch paper. Stray thoughts are not engraved in brass. Sketches on the back of an envelope can be crumpled and discarded. Papers can be shuffled across the desk to endlessly arrange and rearrange a sequence of ideas.

In contrast, a compiled language more closely resembles a bound laboratory notebook in which ideas are more difficult to change once entered.

The power of an idea similar to Visi-Calc might never have been conceived by a professional programmer simply because its structure encourages practices like trial-and-error experimentation which are anathema in a professional application such as development of a company payroll. Someone who earns a living through program development may also have a greater tolerance for non-visible operations and delayed feedback of results.

The needs of the professional and the casual user are not synonymous. Further, the needs of the casual user may be non-obvious to a computer scientist blinded by different needs and more intensive exposure. Only the obvious commercial value of Visi-Calc and variant electronic spreadsheets clearly demonstrated the need for such a tool.

Adults may also have a need for programming languages which encourage experimentation without penalizing it. Such a language would necessarily be both interactive and modular. (Non-modular interactive languages encourage experimentation but quickly produce incomprehensible programs unless prohibitions alien to the natural habits of the novice programmer are imposed by outright force.)

The need for a tool with these characteristics has not yet been generally acknowledged except in some corners of the artificial intelligence community. The demand for such a tool for casual users has been fairly subdued thus far. The lack of resounding cries across the land notwithstanding, concepts such as Visi-Calc are sometimes obvious only in retrospect.

Currently, no interactive, modular computing language is accessible to first-time users across a broad range of different microcomputers. LOGO has these characteristics, but its full capabilities are available only on a few microcomputers at present (such as the Texas Instruments 99/4 and the Apple II).

The increasing number of individuals who are aware of LOGO tend to think of it as a children's language because of its powerful applications in education. In retrospect it may be obvious that adults also have a need for languages for finding solutions, as well as for languages for implementing solutions. ▷

Glen Bull is a professor at the University of Virginia, and teaches LOGO courses at both the graduate and undergraduate level.

LOGO Quilting Party

Have you ever made a quilt? Neither have I. Not a real one, that is. But I have made a LOGO quilt! You and your children can too, by taking advantage of LOGO's ability to piece things together easily.

Remember the now-classic procedure HOUSE? It pieced together the procedures SQUARE and TRIANGLE. And that is just the way a quilt is made: by piecing things together in a set pattern.

This month's activity should help to give your students a better understanding of how complex computer programs are generated: by piecing together simpler ones.

Let's start with a popular quilt pattern called Dutchman's Puzzle. It is made from triangle-shaped pieces of cloth sewed together. Let's learn how to cut out some of these pieces first.

The listing below is a simple procedure to cut out (draw) a patch of the correct shape, that is of a triangle with two equal sides forming a right angle. We will need to know also that the ratio of the length of the long side (the hypotenuse) to the short sides is 1.414, or more exactly, the square root of 2.

```
TO TRIANGLE :SIZE
    MAKE "LEG :SIZE/1.414
    FORWARD :SIZE
    RIGHT 135
    FORWARD :LEG
    RIGHT 90
    FORWARD :LEG
    RIGHT 135
    END
```

Experiment with this procedure a little. Try several different sizes and turtle headings. Make up some of your own patterns with it, using the REPEAT commands and a turn or two.

A quadrant of a patch of the Dutchman's Puzzle has as its main pattern two of these triangular patches joined together with smaller patches of the background color.

Let us concentrate on the two main triangles. The tip of one is placed at the center of the long side of the other. MOVE is a handy procedure to help with this.

```
TO MOVE :SIZE
    PENUP
    RIGHT 90
    FORWARD :SIZE/2
    LEFT 90
    PENDOWN
    END
```

Now we can "sew" a quadrant of our quilt with the procedure QUAD.

```
TO QUAD :SIZE
    TRIANGLE :SIZE
    MOVE :SIZE
    TRIANGLE :SIZE
    RIGHT 180
    MOVE :SIZE
    RIGHT 180
    END
```

Note that we brought the turtle back to its original position and heading. This is very important to do in most repetitive drawing situations you will encounter.

At this point, LOGO's power to synthesize comes into play. Once your children understand how to assemble procedures in this manner, they will be becoming better problem solvers. After all, just about any problem, regardless of its complexity, can be subdivided into smaller less complex subproblems.

Once the subproblems have been divided down enough to become manageable, they can be dealt with one at a time. These "subsolutions" then can be reassembled together to form the eventual complete solution. Making the quilt is a lot like that.

It is easy now to "sew" the first patch.

```
TO PATCH :SIZE
    REPEAT 4 [QUAD :SIZE RT 90]
    END
```

From here, can you figure out the way to assemble a row of PATCHes together? Right! You need another move of some kind. We have called ours HOP.

```
TO HOP :SIZE
    PENUP
    RIGHT 90
    FORWARD 2 * :SIZE
    LEFT 90
    PENDOWN
    END

TO ROW  :SIZE  :NUMBER
    REPEAT  :NUMBER  [  PATCH
:SIZE HOP :SIZE ]
    END
```

And finally, to make the quilt, we need only specify how many rows and figure out how to connect them together. NEXTROW is the procedure we used.

```
TO NEXTROW :SIZE  :NUMBER
    PENUP
    LEFT 90
    FORWARD [ 2 * :NUMBER *
:SIZE ]
    LEFT 90
    FORWARD 2*:SIZE
    RIGHT 180
    PENDOWN
    END
```

continued on page 8

Tipps continued

Teachers will be challenged to differentiate children who can type in a recipe for a circle from those who can both formulate and manipulate CIRCLE for their needs. Need and understanding are the determiners of what to do and how fast to move with the children.

In only two weeks with LOGO, several of the teachers have found that the children are more facile with controlling the turtle than we had supposed they would be. Children are keeping logs of their computer time and notebooks of their progress for assessment.

Teachers will have to make a schedule which suits their class progress. Because LOGO is not a lock-step curriculum now (and we hope it remains so as we learn more about using it), much freedom and responsibility resides with the classroom teacher.

CONCLUSIONS

Every child should have the opportunity to work on the computer and progress toward improved ability in turtling. The teacher provides direction, time, and support. As in direction, time, and support. As in all learning, the responsibility for learning belongs to the learner.

One last guideline in a LOGO environment is that working on the computer is not a frill or extra which can be withheld or denied any more than a teacher would withhold spelling books from a child who talked out of turn or ran in the hall.

Responsible purposeful behavior is expected from all children who work with the computer and may include excitement and fun, but not silliness or carelessness.

On a recent visit to a computer awareness class, I found about thirty "extremely active" students in a converted storage room with five computers. The teacher warned me to not put down the computer I brought because she could not guarantee its safety!

I do not believe that this situation is typical, but I do believe that the many ways that computer education can go wrong are frightening. When computers are treated as something outside the traditional goals of education (nurturing competent responsible human beings), we have taken a dangerous step. As we are just getting started with LOGO, let's not lose sight of the big goals. ▷

Steve Tipps is a professor at the University of Virginia, and conducts LOGO workshops for teachers throughout the eastern United States.

Quilt continued

Now to sew the whole thing together!

```
TO QUILT :SIZE :NUMBER :ROWS
    REPEAT :ROWS [ ROW :SIZE
:NUMBER NEXTROW :SIZE :NUMBER ]
    END
```

In the August (LOGO) issue of BYTE magazine (pages 106-110), Harold Abelson outlined a method to establish quiltlike patterns using the power of LOGO's ability to nest procedures. If you have advanced LOGO students, this would be an excellent time to introduce them to this stimulating concept.

This LOGO exploration could also be used in conjunction with a social studies unit on early American history, or in a home economics class as part of a quilt design unit.

For ideas of other beautiful quilt patterns to adapt for your own use, I recommend Maggie Malone's 1001 Patchwork Designs (Sterling Pub.) and The Patchwork Quilt Design & Coloring Book by Judy Larson and Carol Gull (New Century Pub.). I would like to thank Paula Lewis of Charlottesville for her helpful consultation. ▷

Abelson continued

Perhaps it was not anticipated that the book would receive the heavy use it is getting. The spiral binding and cardboard covers should be replaced with more durable material. But, this is more of a testament to the worth of the book rather than to a publisher's oversight.

If you have an Apple computer and a LOGO version, do not fail to obtain a copy of this book! Harold Ablson, Apple LOGO (for the LCSI version) and LOGO for the Apple II (for the Terrapin/Krell version), BYTE McGraw-Hill, 1982. ▷

From the NLX Mailbag

We have built on the CONTROL procedure mentioned in the November NLX. The TI LOGO listing below has a few more things added, and has the advantage of not stopping the program if a different key (other than E,S,D,X) is accidentally pressed.

```
TO GUIDE
    MAKE "KEY READCHAR
    IF :KEY = "F THEN THAW
    IF :KEY = "F THEN FREEZE
    IF :KEY = "O THEN SY 97
    IF  :KEY = "P THEN PRINT XCOR
PRINT YCOR
    IF :KEY = "H THEN HOME
    IF :KEY = "E THEN SH 0
    IF :KEY = "S THEN SH 270
    IF :KEY = "D THEN SH 90
    IF :KEY = "X THEN SH 180
    GUIDE
    END
```

Elaine Blitman
Punahou School, Honolulu HI

The National LOGO Exchange

ISSN 0734-1717

Volume 1 Number 5 FORWARD 100! January 1983

YPLA Releases Turtle's Sourcebook

To say that The Turtle's Sourcebook is a practical guide to LOGO in the classroom is to understate the case. Authors Jim Muller and Donna Bearden of the Young Peoples' LOGO Association, and Kathleen Martin of the University of Dallas have put together an excellent book of over 100 pages of concepts, background information, activities (both on-computer and off), worksheets, and a list of references for further study.

It features detailed instructions throughout for all versions of LOGO now available, thus giving the reader what is needed, in addition to a feel for the differences between each version.

But do not get the idea that it is a step-by-step one-two-three book. Not at all. To do so would be to defeat one of LOGO's purposes: to give children the opportunity to take charge of their own learning. Instead, the authors give specific suggestions on how to foster an attitude of exploration in the children. "The element of discovery cannot be over-emphasized," they point out.

After an introduction to computers and programs in general, the turtle graphics capabilities of LOGO are explored. Teaching turtle control is emphasized with several suggestions for movement activities.

Next come procedures and several interesting activities and worksheets. Variables are then added to increase the level of sophistication. Recursion is perhaps the most powerful concept introduced.

The authors then give many ideas for putting everything together to construct more complex projects. Testing for conditions and incrementing variables add to the fun. The worksheets in this section are particularly interesting.

The last chapter deals with the list processing capabilities of LOGO. Several of the word and list operations are covered, with many examples. Complete procedure listings of a hangman game and a toolkit of list processing utility procedures is included.

continued on page 8

Turtle Folders Help Third Graders

by

Bonnie Rifkin

Third graders certainly can be LOGO lovers. I'd like to share some of the successful techniques as well as problem areas I have experienced wih this age group.

The particular students of whom I speak are third graders at a private school in Cambridge, Massachusetts, learning Apple LOGO.

My time with the students is very limited: two one-half hour sessions per week. When I am not there, the students do have access to their disks and the computers for additional experimentation. I had to make the LOGO time quality time.

Each student has his or her own turtle folder. The first page of the folder consists of a list of what the student can do in LOGO (draw a square, write a procedure, etc.). Some skills can be completed "alone" and some "with help" and checked off accordingly. This is a constantly growing list and a ready reminder to the students that they are progressing.

Also in the folder is a daily note of encouragement and praise from me, summarizing what was accomplished in the last session and commenting on problems and successes. All worksheets and written procedures are also kept in the folder.

Before students begin their session for the day, folders are passed out. Students look for additional suggestions in their checklists and read their notes. The folders have become very personal to each child and are a good means of positive reinforcement.

I also have found task cards to be most helpful. Even though I deal with no more than two students at a time, these cards could be a life saver to a teacher with a "normal" situation involving many children all at once.

continued on page 2

The National LOGO Exchange, copyright 1983 Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly September through May, $25 per year, $45 for two years, mailed FIRST CLASS from Charlottesville, VA. Add $5 per year for addresses outside the United States. Single copy price $3. Opinions expressed by the authors are not necessarily those of The National LOGO Exchange. Address all editorial correspondence to the editor at The National LOGO Exchange, Post Office Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor......................Tom Lough

What A Time!

I have just returned from the MECC '82 conference in Minneapolis. What a time! It was uplifting to encounter so many upbeat educators all together in one place!

The theme of the conference was "Sharing a Decade of Experience." MECC has been operating in Minnesota since 1972, and has accumulated a wealth of experience and information. They have developed a respected line of software, and have caused Minnesota to be recognized as one of the leading states in the field of educational computing.

They shared their experiences generously, and provided many worthwhile activities, including addresses by Steve Jobs and Alan Kay.

LOGO caused quite a stir! Every LOGO workshop, every LOGO presentation was filled to capacity! In fact, Marcia Horn's last session required opening up an additional adjacent room -- and it was still standing room only! The word about LOGO's incredible potential and the quality of Marcia's presentation had spread like wildfire.

If you have ever doubted whether you are doing the right thing by getting involved with LOGO, you would have become convinced by that crowd!

The experiences I had at MECC will influence me the rest of my life. I heartily recommend that each one of you take time, make time, to go to a computing conference.

Here is one coming up which promises to be a real knockout: the National Educational Computing Conference to be held in Baltimore, Maryland, June 6, 1983. Billed as NECC '83, it has many interesting activities scheduled.

For further information, write to Doris Lidtke, Towson State University, Baltimore MD 21204.

FORWARD 100! ▷

Turtle Folder continued

My set of cards begins with very easy activities, such as "Find out how many steps the turtle takes to go across the screen." and "How many turtle steps are there in an inch?"

Other cards give suggestions and challenges for ideas of simple figures to try to reproduce or alter, such as stars, triangles, and sketches of more complex figures, such as a tree.

The task cards help in several ways: (1) For the child who never knows what to draw, is unmotivated, or needs structure, the cards give a specific activity and take away the anxiety of "You have twenty minutes. What can you draw on the screen?" (2) Developing task cards forces the teacher to formulate a sequence in his/her mind. (3) The cards can be used as a kind of worksheet in situations requiring the teacher to work exclusively with one student while the others remain purposefully occupied. (4) Most importantly, the activities can be springboards that will give students ideas for further personal projects.

Additionally, I have noticed that children of this age group often have difficulty visualizing on paper where the turtle will go. Thus, I initially encourage my students to plan their projects right on the computer in the immediate mode.

The first time I did this, the figures they were designing eventually appeared on the screen. But when the children discovered that their drawings could not be saved directly (they were not yet procedures), frustration and tears followed.

Then we realized that, if they worked in pairs, with one writing down the commands that the other typed in, things went better. After they got the drawing they wanted, they simply went back and wrote it as a procedure, copying from what the partner had recorded.

When they decided they did not want to enter their commands twice, they realized the value of planning on paper.

In summary, turtle folders and task cards have been helpful to me. I am still working on ways to help my third graders plan projects. ▷

Bonnie Rifkin is an educational consultant, teacher, and an intern at the Technical Education Research Center (TERC) in Cambridge, MA.

TIPPS for TEACHERS

by

Steve Tipps

From Turtling to Programming

In the first phases of working with LOGO, learners control turtle movements and discover the turtle world. Beginning LOGO-ers explore the turtle's environment using the distance (FD, BK) and direction (RT,LT) instructions. The turtling explorations become the foundations for programming.

Perhaps "programming" actually begins with the first turtle movement; control is certainly the primary idea in programming. Programmming, however, suggests multiple actions performed in a delayed manner often utilizing some efficient shortcuts.

Indirectness is the best general description for programming. Turtling is usually done with only a few actions performed immediately and specifically. Turtling in the immediate mode builds understandings important in programming the turtle for more interesting complicated actions.

Helping children move from turtling to programming is the issue raised in a letter from Steve Matthias, principal of Washington Center Elementary School in Fort Wayne, Indiana. His question may be the most crucial for all who hope that LOGO will fulfill its Papertian promise. In response to Steve's question, I'll share two examples of guiding students toward programming.

The students in a middle school exploratory class were familiar with turtle commands, pen and background colors, and REPEAT. Working with REPEAT serves as an introduction to a shortcut way of executing a sequence of actions. To facilitate working with REPEAT, the teacher had also introduced a procedure using TO (space) to enter the editor and CTRL-C to define. Although all of the children had mechanical knowledge about making procedures, none had used procedures in very purposeful ways. They were still turtling around the screen and making colors change.

WILL'S STOP SIGN

While exploring REPEAT, the students made squares, circles, and other regular polygons. One boy, Will, drew an octagon which, in turn, inspired him to draw a stop sign. He worked for two days on writing S, T, O, and P within the octagon. Only OCTAGON was being drawn as a procedure. An indecipherable string of FD, BK, LT, RT, PU and PD was used to complete the figure. All of these actions were performed in the immediate mode; trial and error finally brought about reasonable sizes and positions.

The question was posed whether other parts of the picture had names which could be defined as procedures. No response to the question came at first. But, on the next stop sign, he said, "The first thing I need is an S, then a T, then an O, then a P." In naming the parts of the project, Will had changed the nature of his effort and had moved appreciably toward programming.

Will went to the editor with TO S, described S and defined it, returned to graphics, drew OCTAGON, drew S, and found that he had forgotten to position it! He returned TO S, included positioning moves, returned to graphics, drew OCTAGON and S satisfactorily, and then went on to work out T, O, and P. By moving back and forth between graphics and the procedure editor, he got his stop sign.

Will had five procedures which he could name separately to form his design. The teacher then asked if he could teach the turtle how to do all five things with one instruction. After a few moments, Will responded that he could teach the turtle how TO STOPSIGN, and created a master procedure:

```
TO STOPSIGN
   OCTAGON
   S
   T
   O
   P
   HIDETURTLE
   END
```

The teacher introduced SAVE "WILL.STOPSIGN to record the file on disk. The next day, Will READ the file, called his procedures and began to use them in new and different ways. The T procedure used in a REPEAT instruction made a kind of railroad track all over the screen. O and P were rotated and created some interesting effects. S did not do much. But each of the parts of STOPSIGN lead to new explorations.

The trial and error decisions which had been made were important to the completion of the project. Letter sizes and positions were already developed; had they not been, the process would have been important to complete concurrently. Will had defined a project, worked through a solution, and begun to build a repertoire of procedures to explore and elaborate upon.

continued on page 4

Tipps for Teachers continued

DAVID's ZIG AND STAR

On another day, the teacher had proposed the task of making zigzags. This is an exercise which I consider excellent for consolidating distance and direction control with a REPEAT statement. Several observations lead me to this recommendation. First, making regular polygons is so familiar to children that doing it does not demand full understanding of distance and direction. Second, regular polygons use only LEFT or RIGHT while zigzag designs require both. Finally, zigzags are so variable that they generate many projects.

The students were confident enough with the screen turtle that they immediately began putting things into REPEAT's. A number of designs resulted, but neither zigs nor zags. As the teacher moved from group to group, suggestions were made to draw or walk through a zigzag and record the instructions for exactly what happened. As students talked through the figure, they found a pattern of FD, turn, FD, turn, with the turns being in different directions. Some of the groups took the pattern immediately to a REPEAT while others did the pattern step by step.

After isolating the pattern, the groups went different ways. Directions and distances were not equal in all the patterns, resulting in many different zigzags. David settled on a zig pattern which was wide-angled. His zig was composed of:

```
FD 20 LT 75 FD 20 RT 75
```

After typing the pattern into repeat statements several times, David was challenged by the teacher to make a procedure having only those commands in it. With reluctance, David defined TO ZIG. He simply saw no purpose to doing it.

But when he returned to immediate mode and began experimenting with ZIG in REPEAT statements, he soon found how much easier having ZIG was than retyping even four simple commands again and again. He explored several things before stumbling onto:

```
ZIG RT 75
ZIG RT 75
ZIG RT 75
ZIG RT 75
ZIG RT 75
```

which made a star with one point crossed rather than pointed. The RIGHT 75 came from the angle in ZIG rather than any particular plan. The pattern David found led to the command

```
REPEAT 5 [ ZIG RT 75 ]
```

then, to adjust the angle,

```
REPEAT 5 [ ZIG RT 70 ]
```

and finally to

```
REPEAT 5 [ZIG RT 72 ]
```

which caused a star to be drawn.

The teacher suggested that David might waant to make a STAR procedure. He did this without as much reluctance! The star was then drawn and rotated and moved about the screen. One of the most impressive displays was

```
REPEAT 40 [ STAR RT 9 ]
```

When David decided that the star was not big enough, he first tried to change STAR, then ZIG. He had recognized that the size was controlled by the FD command and that FD was in ZIG rather than STAR. He changed the size of STAR several times by editing ZIG.

MOVING FROM TURTLING TO PROGRAMMING

The vignettes are offered both because they are excellent examples of guiding children toward programming. A number of people would find too much intervention in the teacher's approach. The examples are offered instead because they are realistic portrayals of a teacher's struggle not to infringe on exploration and also to move the children toward some power with LOGO.

Students get bored or frustrated if they don't have enough power to do more than random turtling. Defining procedures which become new objects gives learners more to explore. The mechanics of REPEAT and of defining procedures are not as important as the kind of thinking which is encouraged by appropriate use of them. The teacher asked questions and made suggestions aimed at improving problem solving with programming. Some of the strategies being developed were:

1. Moving back and forth from immediate and editing mode. After students learn about editing, they tend to start by defining a procedure which they expect to work automatically rather than working things out in graphics first. (See Bonnie Rifkin's article in this issue.) The teacher also reinforced body syntonicity and drawn models for working out ideas.

2. Defining small and understandable segments of code. OCTAGON, S, T, O, P, STOPSIGN, ZIG, and STAR are coherent sets of instructions.

continued on page 6

MICROWORLDS

by

Glen Bull

How to Use LOGO

Some of the characteristics of LOGO have been discussed in previous columns. The interactive, modular nature of LOGO makes it particularly suited for the introduction of programming concepts. An interactive language provides faster feedback, which accelerates the learning curve.

A modular language makes it possible to break large programming problems into smaller parts. In addition, LOGO has a number of other features which makes code generally more readable. That is not to say that the same sorts of things can not be accomplished in other programming languages; just that it is more convenient in LOGO.

PROGRAMMING CONCEPTS

The suitability of LOGO for introduction of programming concepts creates a temptation to use it for just that. Some predictions suggest that many of the smokestack industries such as steel will be lost or substantially reduced by the end of the decade.

If so, the middle of the job market could drop out, leaving the unskilled jobs at the lower end and a plethora of technological positions at the other end. The result might be the curious phenomenon of numerous unfulled jobs coexisting with high unemployment.

Given this forecast, the expense of introducing LOGO in the schools might be justified on the basis of economic self-preservation. A student who can program with facility in LOGO, using its advanced capabilities such as list processing, should be able to transfer the same concepts to most other common programming languages. A student who can program is also likely to have an increased probability of finding a job.

It is true that the programming position will not necessarily require knowledge of LOGO, but neither will it necessarily require knowledge of other languages commonly used to introduce programming. For example, many professional programming positions involve use of COBOL, a language which is not commonly used as an introduction to programming - nor should it be. Acquisition is more important than knowledge of the mechanics of a particular language.

Having noted this, I must confess that I am dubious about the programming rationale as a justification for getting LOGO into the schools. The economic arguments are strong, and are likely to become stronger by the end of the decade.

However, carried to extreme cases, I have nightmarish visions of elementary teachers racing to see which one can teach the most LOGO commands by the end of the school year. A rigid curriculum for the "proper" way to introduce these commands would be developed. Teachers and children would be forced to march in lock-step fashion through a specified content at each grade level. In short, LOGO could be used for the exact opposite of the purpose for which it was developed.

LOTS OF OTHER USES

There are a number of uses to which LOGO can be put, in addition to its potential value as an introduction to programming concepts. I recently reviewed several elementary school science texts. A typical chapter included a picture of the center of the earth with concentric circles of brown, yellow, and red. As I recall, when I was a child at that grade level, my class constructed clay volcanoes with baking soda in the middle to simulate an eruption.

The difficulty with such an approach to science is that it is authoritarian. The child is told to believe the center of the earth is molten because the teacher or an even more distant author says so. There is no way for the child to personally verify the truth of the statements made. Experiments may be trivial, or possibly preselected to demonstrate only evidence favorable to the interpretation selected. For example, I suspect that most chidlren are aware that baking soda in a well of clay proves very little about the contents of the center of the earth. It is fun, but also irrelevant.

The most powerful aspect of the scientific method is the effective means it provides for thinking about how the world works. An important component of this method is hypothesis testing and personal verification. The difficulty with the textbooks I reviewed was the limited number of opportunities for children to convince themselves that these methods work.

Part of the reason for this lack of opportunities is economic. The equipment required for children to engage in personal verification would be too expensive. The time required to supervise each child would also be prohibitive, since true science involves quite a bit of backtracking and false trails. I suspect that even

continued on page 6

Microworlds continued

if time and expense were no object, specialized "science education" equipment would be developed which would dictate prescribed paths each child was to take.

In this educational environment, LOGO can be used to validate the teacher's authority. There are certain principles which are proven problem-solving strategies. There are a limited number of opportunities in a classroom environment for children to demonstrate to themselves that these principles work.

EXPLORATION

A task in LOGO selected by the child precludes the possibility that the teacher conspired to select the single example in the universe which will yield a particular set of results. The child can try strategies suggested by the teacher and compare them with approaches suggested by other children.

The computer system acts as an impartial arbiter in a democracy of ideas. An incorrect algorithm will not run solely because of an assertion that it should work. The fact that teachers are occasionally incorrect about problem solving strategies only lends credence to the majority of times in which suggested avenues prove fruitful.

One difficulty with using LOGO in this way is the problem of access. The child needs time to fumble around and convince herself that an inefficient strategy will not work. This may cease to be a problem as families opt for a computer in place of an encyclopedia.

Another difficulty is the higher level of skill required for this approach. Anyone can march a group of children through a prescribed sequence of programming instructions. No special skills are required - just follow the manual and grade the exercises provided. Do not allow the students to work on unauthorized projects, or else limit this type of work to special unsupervised activities by brighter students.

Teachers who allow children to use LOGO to convince themselves that certain styles of thinking are effective will come closer to creating true scientists. The fact that these children will gain a solid grasp of programming concepts along the way will be a mere side effect of the process. ▷

Glen Bull is a professor at the University of Virginia, and teaches LOGO courses at both the graduate and undergraduate level.

Tipps continued

3. Naming of the sets of code with meaningful labels. Each of the names is descriptive of the procedure. This is a feature which sets LOGO apart from most other languages.

4. Using procedures in many different ways. Combinations of procedures with other procedures and commands emphasize the modular and independent nature of procedures, encouraging problem solving in small chunks.

The teacher did NOT do several things. First, although the "magic" number 360 appeared several times in the course of the projects, the teacher did not stop and give a lecture about the Total Turtle Trip Theorem. During STAR, the teacher asked what 5 times 72 was and got the answer 360. Questions about SQUARE and CIRCLE also got the answer 360, but no recognition that this was something which could be used for planning.

Second, the teacher did not dissuade the students from putting positioning moves inside the procedures. Although not a good practice in the long run, at this time the students did not seem ready to contend with the complication. Need for generalizable procedures will come with new projects. For instance, spelling POTS or SPOT with S, T, O, and P would be possible but not neat. The teacher might suggest these words, knowing the problems encountered would point to problems of positioning within procedures.

I often ask students if PC and BG should be put into procedures or left out. The discussion leads to the determination that there are no absolute answers -- only situational ones. "If you put the color inside the procedure, it will always be that color." The issue of position is analogous. Eventually, recognition of what you always want to happen and what you want to be able to set will support the idea of variables.

Finally, the teacher did not introduce variables, even though changing the star size would have been much easier with variables. Variables provide much power; but many things are possible with the power at hand. LOGO does not need to be approached in a rapid lock-step fashion. Exploration at the level of procedures can be very challenging. Not only that, doing things in a clumsy way makes the short cut more understandable and appreciated.

continued on page 8

Can the Turtle Draw a Sine Wave?

It is fun to give presentations on LOGO to teachers and other interested people. Most of the time, everyone gets very excited and a bit awestruck when they realize the power of the language.

Occasionally, a member of the audience will let the imagination soar and think of a completely unexpected application for LOGO.

That happened to me recently. After my presentation, I was taking questions about what had just been demonstrated. Then up went the hand!

"Can the turtle draw a sine wave?"

Everyone got quiet and looked at me. Hem. Haw. To tell the truth, that had never entered my mind. So, let's follow our own advice!

"I don't know," I said. "But I think it could. That certainly is an intriguing question!"

I couldn't get the idea out of my mind. Later, after I was able to think about it some more, a few ways to start came to me.

How would I walk in a sine wave? Well, it is just going forward a little and turning a little within a certain range. But how much to turn?

From my etch-a-sketch days, I recalled that I made something like a sine wave by rotating one knob at a constant rate while I "rocked" the other back and forth. In fact, that rocking was really a periodic input, just like the sine wave output.

Then I began to understand. The turtle would have to turn according to the derivative of the sine wave! (The derivative is a measure of the rate of change of a function, and is an important part of the calculus.)

This came as a surprise, but, then again, the more I explore with LOGO, the more depth I discover. Lots of other people are reporting similar experiences.

Since the listings furnished are far from perfect, why don't you try to develop your own independently?

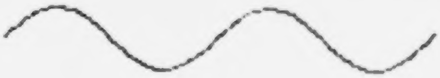
The EXPANSION in SINE is just a scaling factor which controls the size. I had wanted to be able to control both the height and the width separately. But I have not yet gotten a good grasp on all the scaling concepts. EXPANSION controls both the height and width in these listings.

STEP is the amount the angle is changed for each point on the sine curve. I found a good compromise between speed and smoothness with values between 5 and 10 degrees.

the turtle is turned to an initial heading of COS :STEP. Then, in the SKETCH procedure, it goes forward an amount determined by the angle and the values of EXPANSION and ANGLESTEP. This amount is not strictly correct, but it gives a nice result.

Then the angle is increased and the turtle is turned as a function of the cosine of the angle. (The cosine is the derivative of the sine.) Note the need for the absolute value of the cosine. Can you figure out why? Try it without using ABS!

Here is what SINE .1 5 looks like:



You can do much better than this listing, I'm sure! If the "sine bug" bites you and you work up a procedure which is satisfying, send it in to share with everyone! Good luck!

MIT VERSION

```
      TO ABS :VALUE
           IF :VALUE<0 OUTPUT -:VALUE
           OUTPUT :VALUE
           END

      TO SINE :EXPANSION :STEP
           DRAW
           RT ATAN COS :STEP 1
           PU SETX -100 PD
           MAKE "ANGLE 0
           SKETCH :EXPANSION :STEP
           END

      TO SKETCH :EXPANSION :STEP
           FD :EXPANSION * :STEP *
3.14159 * ABS COS :ANGLE / 180
           MAKE "ANGLE :ANGLE + :STEP
           RT ((90 - HEADING) - ATAN
COS :ANGLE 1 )
           SKETCH :EXPANSION :STEP
           END
```

For the LCSI version, put brackets around OP -:VALUE in ABS, change DRAW to CS, and change ATAN COS :ANGLE (or :STEP) 1 to ARCTAN :ANGLE (or :STEP).

FORWARD 100! ▷

Tipps continued

Projects are the motivation for programming: having something you're interested in doing is essential. Turtle graphics provide a set of projects for children and adults other than accounting and engineering ones. The teacher gave the children a lot of freedom in deciding projects. The ZIGZAG exercises are more skill builders than a single project per se. Both STAR and STOPSIGN came from exploration.

Guiding chidren to programming allows them to complete their projects with skills which grow with them toward solution of more complex problems. Guidance has alays been the most challenging teaching role. Finding the right combination of intervention (See "Hold Your Horses" November NLX) and benign neglect will be the main task for LOGO teachers -- and the most fun. ▷

Steve Tipps is a professor at the University of Virginia, and conducts LOGO workshops for teachers throughout the eastern United States.

Turtle's Sourcebook continued

Lastly, three appendices include a treatment about using the edit mode, a LOGO resource list, and a look at computer languages.

If you have been looking for a teacher-oriented guide to introducing LOGO to your classroom, we recommend The Turtle's Sourcebook.

The book, 20 worksheets ready for duplication, and a demonstration disk in the appropriate LOGO version come packaged together for $29.95, plus $3.00 shipping and handling.

Mail your orders to Young Peoples' LOGO Association, 1208 Hillsdale Drive, Richardson, TX 75081. Be sure to specify your LOGO version. ▷

Get Your NLX ABC's!

Many NLX-ers have received their LOGO ABC's already. In case you missed the invitation in the December NLX on page 2, to get your listing of how to draw all the ABC's with the Apple computer and any of the LOGO versions (please specify), send a self-addressed stamped envelope to NLX ABC's, Box 5341, Charlottesville, VA 22905.

Learning with — — — — LOGO

Have you ordered your copy of Dan Watt's new LOGO book from McGraw-Hill yet? If so, you may want to check your order.

We have been informed recently that there will be two separate books published, in a manner similar to those of Abelson's (See December NLX). One title, "Learning with LOGO," has appeared in many notices to date. However, what the notices did not say was that this is the title for the MIT (Terrapin/Krell) version of LOGO. If you have the LCSI LOGO (Apple LOGO), you should order the second title, "Learning with Apple LOGO."

The LCSI title has not received much publicity. This may cause some inconvenience to those of you with LCSI who have already ordered "Learning with LOGO" in advance of the publication date. If you act immediately, it may be possible to change your order before you receive the wrong book.

Release of "Learning with LOGO" is scheduled for sometime in January, 1983. According to information given us, "Learning with Apple LOGO" will not be released until two weeks later.

Many LOGO people are waiting eagerly for Watt's books. His writing style is excellent and his reputation in the LOGO field is outstanding. We may have another LOGO classic on our hands very soon.

LOGO to Pascal?

Two educators recently announced a plan to consider teaching a high school course in Pascal. However, they will be teaching their students LOGO for the first six weeks! They hope to investigate whether LOGO will provide an easy entry into Pascal, a high threshold language.

The course will be general in nature, and will not necessarily be designed for advanced placement candidates.

They would like to hear from other interested educators, particularly those with both Pascal and LOGO experience.

Write to:

Robert D. Nelson
Richfield Public Schools
70th St. & Harriet Ave. South
Richfield, MN 55423

Charles Schlimpert
Lutheran High School
2222 North Santiago Boulevard
Orange, CA 92667

The National LOGO Exchange

ISSN 0734-1717

Volume 1 Number 6 FORWARD 100! February 1983

Where Does LOGO Fit In?

by

Bonnie Rifkin

Several years ago, school systems began programs for gifted students. These programs tended to be set up (and still are) so that students would leave their regular class for a period of time and either go to a resource room or get on a bus and go to a center for their differentiated learning.

One of the fears with such a set up was that the gifted program would become an entity unto itself, cut off from and in no way relating to the rest of the day's classes. In the majority of cases that fear has been overcome.

The point that I am trying to make is this: as computers make their way into the schools, we must not let them fall prey to the same fear. Just like gifted education, computers are here to stay.

In schools where the gifted students study LOGO, I feel strongly that their experiences should be brought back to the regular classroom and incorporated into the other subjects.

Sharing with classmates is very important. Gifted students often become very excited about what they have done. Why not provide a structured time set aside for them to share their experiences?

Students may explain a new procedure, demonstrate it, or describe how they went about debugging it. Knowing ahead of time that this special period exists, students may even provide visual aids as a part of their LOGO presentation. The activity then doubles as training for presenting in front of a group.

In the case where there are several students that may be feeling frustrated with LOGO, a group project is the perfect solution. Have them pick a theme for a mural, for example. The theme might relate to a science or social studies unit. Split the class into four groups. (For a large class create multiple murals to keep groups small.)

continued on page 7

LOGO and the Single Computer: How to Cope

by

Linda Nix

There has been much excitement in my classroom the past few weeks as we worked with LOGO on the TI 99/4 computer.

Students were eager to come before and after school in order to have extra time to experiment and explore. Several students read ahead in the manual to learn new ways to create programs and then shared their finds with other enthusiastic students. Students excitedly bragged about how they were able to "program" the computer. Students from other grades came before and after school to watch. As a teacher, I, too, was excited because I had become a facilitator rather than just a demonstrator.

The major problem became how to provide time on the computer for everyone who wanted hands-on experience and still work with the established math curriculum. I've tried two different methods so far. Both have been fairly successful. This article's purpose is to share these ideas and possibly stimulate new ideas for time management where only one computer is available.

TWO AND TWENTY

As we began working in the turtle mode, we divided the students into pairs. They took turns acting as secretary (kept records on paper) and programmer (typed in directions). The class period was divided into three twenty-minute sessions and all twenty students were able to use the computer at least three times during a two week period. Students needing more time came before and after school.

The computer was placed in the back of the room. While each group worked, the remainder of the class participated in drill and practice math activities. This gave me time to work with individual students needing help.

continued on page 2

The National LOGO Exchange, copyright 1983 Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly September through May, $25 per year, mailed FIRST CLASS from Charlottesville, VA. Add $5 per year for addresses outside the USA. Single copy price $3. The opinions expressed by the authors are not necessarily those of The National LOGO Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress St., Salem, MA 01970. Address all other correspondence to The National LOGO Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717/83 $2.00

Editor......................Tom Lough

RIGHT 15!

From the spirit of your letters and cards, we are getting indications that the NLX has met many of your needs thus far. For this, we are both pleased and grateful.

However, we would not be serving your best interests if we were not willing to change and grow with you. FORWARD 100, and full steam ahead is great! But it may be time to consider a small change in our direction.

We are all most comfortable doing what we know how to do best. To change and to grow are unpopular ideas with many people. They demand a special effort and some extra time. Yet, it is only by changing and growing that we progress.

In this issue, you will see some new directions, mainly toward the text and list processing part of LOGO. We feel it is time to do this.

Turtle graphics are wonderful! I'm sure there are those who could spend ten years teaching with them and never repeat themselves.

But, unless we help the children discover the added capabilities of LOGO's list processing, aren't we sort of keeping something from them? It is in this part of LOGO that its real power resides. Yes, it can be confusing and complicated. But we are teachers. We can dig it out!

Besides, it seems that the public views LOGO as just turtle graphics. And, why shouldn't they, since almost all the articles have stressed that aspect? Are turtle graphics enough to win the support of the public for LOGO? Perhaps not.

By considering a small change in our direction of LOGO teaching at the right time, we may be turning to a heading with unimagined rewards for you, your children, and your world.

RIGHT 15, FORWARD 100!

Single Computer continued

Most students were able to put their perfected programs in the edit mode by the third session. The edit mode was an excellent way for me to check quickly the accuracy of the student's program without going through the entire thing.

Programs were saved on a disk and recorded in a class LOGO book so students could work each others' programs. This scheduling method worked well, but students did not like the break in sessions before finishing their programs. As they started the second step of programming (this time in the sprite mode), we grouped differently to allow more continuous computer time.

THREE AND SIXTY

Students were divided into groups of three. Each group was given one full class period, approximately 60 minutes, to create and name a sprite program. The computer was put in the hall, so the regular math class could still be conducted without disruption.

When all programs were completed, student groups presented their programs to the entire class. Each group reviewed other groups' programs for debugging and refinements and the programs were stored on a disk.

ALL TOGETHER NOW

For the third programming project students could either work in groups or independently to develop a program which included both the turtle and sprite modes. In most cases they included their own sprites (shapes) in their programs.

Students were scheduled to work on their programs uninterrupted for three entire class periods. This included working out programs on paper, testing them on the computer, and writing up a final copy to turn in. For this project we were able to borrow two additional computers from the high school.

In addition to working on their programs during math class, fifth grade students acted as student teachers for students in first and third grades. These younger students came every day for 45 minutes and were shown the basic LOGO commands. They were then allowed to program the computer. Different fifth grade tutors were used each day so they only missed one or two of their regular classes.

This project was a huge success! The fifth graders worked extremely well with the younger students, thereby gaining confidence and improvement of their computer skills. The younger students had fun and lost their fear of the computer.

continued on page 8

TIPPS for TEACHERS

by

Steve Tipps

The Other Side of LOGO

Turtle graphics have been the subject of my comments and suggestions about teaching with LOGO. The reasons are plain. Turtle graphics are intriguing, creative, compelling, exciting, and form the basis of learning about programming through LOGO. In fact a number of companies market something they call "LOGO" which consists only of turtle graphics. However, turtle graphics is only half (or less) of the power of LOGO.

The other side of LOGO is working and playing with groups or lists of characters such as words, numbers, sentences, and phrases. The list processing capability of LOGO offers new ways for children to learn about language and its rules, just as they learned about geometry and its rules through the turtle. List processing should not be kept on the shelf indefinitely but should be introduced early - along with the turtle graphics. This column suggests some ways which children can prepare to learn about LOGO list processing and use it.

CONTROL AND CORRECTION

Just as children had to know FORWARD, BACK, RIGHT, AND LEFT, a few primitive commands are needed to get started with word play. First, the children need to get the computer to print out messages.

```
        PRINT [ I AM SALLY AND I LIVE IN
SYRACUSE. ]
        PRINT [ THE DAY IS CLOUDY AND
GRAY. ]
        PRINT [ I AM TALKING TO THE
TURTLE.]
```

In the immediate mode, each of the statements will be printed or echoed back to the writer. The writer has the pleasure of seeing what has been written. The computer responds in messages instead of with turtle movements. The editing keys which have been used in turtle graphics (destructive backspace, errors, and delete) can be used to correct any mistakes which are found.

The commands and comments here are directed toward the MIT (Terrapin, Krell) LOGO versions. However, most of them can be used directly with Apple (LCSI) LOGO. The commands for the TI version are similar but not identical.

Remember that <ctrl P> can be used to recall the previous line so that it doesn't have to be retyped every time.

The next step in getting words out of the computer is to write small echo-type programs.

```
        TO INVITATION
           CLEARTEXT
           PRINT [ FEBRUARY 3, 1983 ]
           PRINT [ JAMIE, ]
           PRINT [ I AM GOING TO HAVE A
VALENTINE PARTY. ]
           PRINT [ COULD YOU COME? ]
           PRINT [ IT WILL BE AT MY
HOUSE AFTER SCHOOL. ]
           PRINT [ ]
           PRINT [ YOUR FRIEND, ]
           PRINT [ PAT ]
```

The writer produces a message on the screen or on paper with a printer when INVITATION is called. By writing in this delayed mode, the writer now constructs and stores messages for others instead of writing words for immediate personal use. The writing task has changed. Creating a sense of purpose and a sense of audience are important objectives for such creative writing.

The INVITATION may not be right the first time. Corrections in spelling and punctuation are simple. The friendly editor in LOGO allows movement of the cursor to the next line <ctrl N>, previous line <ctrl P>, first of line <ctrl A>, end of line <ctrl E>, and to open a line <ctrl O>. Screen editing in LOGO may not be as sophisticated as WordStar, but it is better than most program editors.

Additions and deletions are also easy. Someone might notice that the day, date, and time of the party, and Pat's home address have been left out. Each could be added into the procedure without having to start from scratch.

Spacing could be improved with blank lines by the insertion of PRINT [ ] between the date and the salutation. However, INVITATION will still print all of the lines starting at the left margin. To adjust the spacing of the letter, often called formatting the letter, CURSOR moves can be used. The textscreen is composed of 40 spaces across (numbered 0-39) and 24 lines down (numbered 0-23). CURSOR needs two inputs to tell it where to start printing. If the date is to start in the center of the page and 10 lines down, then

```
        CURSOR 20 10
        PRINT   [ FEBRUARY 3, 1983 ]
```

in INVITATION would position the date. The closing might be directly under the date but on line 15 with the signature 3 lines under it:

continued on page 4

```
CURSOR 20 15
PRINT [ SINCERELY, ]
CURSOR 20 18
PRINT [ PAT ]
```

Other page formatting can make the message like a stair step or indented. The important idea in experimentation with PRINT and CURSOR is that the words can be controlled and corrected just as effectively as turtle moves. When children can print out messages in both the immediate mode and the delayed mode, they are ready to move into some more sophisticated uses of LOGO list processing.

AN INVITATION FOR ALL

In addition to controlling printing by editing a program, the writer can also change the message from the keyboard. If Pat wanted to invite 10 friends to the party, ten invitations could be printed with a REPEAT statement. OUTDEV 1 directs the message to a printer assumed to be in slot 1. OUTDEV 0 returns output to monitor.

```
OUTDEV 1
REPEAT 10 [ INVITATION ]
OUTDEV 0
```

Now Pat gets ten invitations, all addressed to friends named Jamie! One further change might be to eliminate the salutation line and write in each name by hand. Or, INVITATION could be edited ten times, changing the name each time. Both of these actions are rather clumsy and do not tap the ability of the computer.

But why not let the children try each of these methods at least once before moving on? I believe that doing things the old-fashioned way builds an awareness of the process which you later want to computerize - rather like walking through turtle commands. I find that having done things the slow or tedious way makes learners less in awe of the process and more appreciative of the product.

The more sophisticated method involves changing the name by using a local variable. Looking into the original INVITATION procedure, the student should be able to see that the line which needs to be changed is PRINT (JAMIE). But, JAMIE is a "value" for a name. What needs to be changed 10 times is the value of NAME, a variable.

The idea of variables may or may not have been introduced formally, although children will have been using the concept since the first time they typed in FORWARD 10. FORWARD is a command which needs a value before it can be executed. From the keyboard we can give FORWARD a value of 10, 20, 100, or anything else. The value can vary or change.

(The debate about how and when to introduce variables continues, but that's an issue which I'm sure Glen Bull will take up in a future column. For now, I'm going to pretend there isn't even such an issue.)

Variable control can be applied to list processing by fixing INVITATION in the first line which states the procedure and in the lines which contain the item to be changed.

```
TO INVITATION :NAME
   .
   .
   PRINT  :NAME
   .
   .
   END
```

The first line says that when INVITATION is called, something else also must be specified as an input. In this case, it is the :NAME of the invitee. The colon tells LOGO you are referring to a variable value rather than a procedure name. Thus, ten INVITATIONs can be printed by:

```
OUTDEV 1
INVITATION "JAMIE
INVITATION "JOHN
INVITATION "DANA
.
.
INVITATION "NATHAN
OUTDEV 0
```

The convention for giving the computer a name to put in INVITATION requires an initial quote mark. The rationale becomes clearer with some of the later list processing capabilities.

The process by which computers automatically insert our names into those all-too-familiar form letters from mailing lists is beginning to emerge. Word processing and file merging are important computing functions which will be used by adults and children. LOGO encourages exploration of such powerful computing ideas by making the process accessible with a simple vocabulary and obvious structure.

Although this example is cumbersome from LOGO's point of view, the ability to specify from the keyboard something inside the program is powerful. List manipulating abilities of LOGO will be treated next month.

WORD PROJECTS

Another classroom project could be composition of poems:

```
    TO MARY.POEM
       PRINT [SOMETIMES I'M HAPPY. ]
       PRINT [ SOMETIMES I'M SAD. ]
       PRINT [ BUT WHEN YOU ARE MY
FRIEND, ]
       PRINT [ I'M ALWAYS SO GLAD. ]
       END
```

continued on page 7

MICROWORLDS

by

Glen Bull

Introducing the Text Screens

When they first begin LOGO, most children use the text screen only as a transition to turtle graphics. However, LOGO also has great potential for activities in the language arts. A number of its instructions are designed to facilitate play with words and sentences.

Many of these activities are dependent on use of recursion. However, stepping stones can be introduced before the concept is treated in detail. Let's examine a few possibilities, using MIT LOGO.

GETTING ACQUAINTED WITH THE TEXT SCREEN

The text screen can be visualized as a grid 40 spaces across and 24 lines down. The spaces are numbered 0 to 39 and the lines are numbered 0 to 23.

The cursor (the blinking square) can be sent to any space or line on the screen. For example, CURSOR 19 1 sends the cursor to a point 20 spaces over in the second line of the screen.

In previous NLX issues, Steve Tipps has noted the importance of familiarization with a concept before applying it at more abstract levels. The user should be allowed to experiment with direct execution of cursor positioning instructions on the text screen before using them in procedures.

Cursor positioning instructions can be combined with PRINT statements to place <u>text</u> at various positions on the screen. Previous text on the screen is cleared by use of the instruction CLEARTEXT.

```
CLEARTEXT CURSOR 19 12 PRINT
[HELLO    THERE!]
```

LOGO normally disregards the extra spaces between words. For example, the line above would be executed as:

```
HELLO THERE!
```

Extra spaces will be retained if the text is placed between single quotation marks. Thus, the line PRINT ['HELLO   THERE!'] will be printed as:

```
HELLO     THERE!
```

A PRINT statement blanks the rest of the spaces on a line beyond the printed text and sends the cursor to the beginning of the next line. For example, REPEAT 4 [PRINT [X]] prints each "X" on a separate line:

```
X
X
X
X
```

In contrast, PRINT1 leaves the cursor at the end of the text printed. Thus, REPEAT 4 [PRINT1 [X]] produces:

```
XXXX
```

These instructions can be executed directly in the text mode without knowledge of any other instruction except the REPEAT statement. Hence, they can be introduced at almost any time after the initial introduction to LOGO. As soon as the user begins to define procedures, these instructions can be combined to produce simple animation effects.

IT MOVES!

Animation can be created by drawing an object, erasing it, and redrawing it in a new place. The object must be erased and redrawn in less than a twentieth of a second to create the illusion of movement. These rates can be achieved with a series of PRINT statements.

In fact, a series of PRINT statements can draw and erase an object so quickly that it may not be seen. Therefore, it is necessary to insert a WAIT statement before erasing the object and redrawing it in a different position.

Other versions of LOGO (LCSI and TI, for example) have a WAIT instruction as a builtin command, or primitive. For versions that do not, it is easy enough to create one.

```
TO WAIT :TIME
   IF :TIME<0 STOP
   WAIT :TIME-1
   END
```

The WAIT statement takes a number supplied to it and repeatedly subtracts one from the number until the number is less then zero. The larger the number, the longer the wait.

To illustrate, an arrow can be constructed from a few dashes and an angled bracket:

```
TO ARROW
   CURSOR 0 10
   PRINT [--->]
   CURSOR 0 10 WAIT 5
   PRINT ['        --->']
   CURSOR 0 10 WAIT 5
   PRINT ['                 --->']
   CURSOR 0 0
   END
```

continued on page 6

Microworlds continued

The rate at which the arrow moves across the screen can be increased or slowed by changing the number after WAIT. Since the arrow appears to hang in mid-air at the end of its flight, a backstop provides it with a reason for stopping.

```
TO BACKSTOP
   CLEARTEXT
   CURSOR 18  9 PRINT1 [!]
   CURSOR 18 10 PRINT1 [!]
   CURSOR 18 11 PRINT1 [!]
   CURSOR 0 0
   END
```

The effect is even more satisfying if an impact is heard as the arrow strikes the backstop. There are a number of special characters which do not appear on the screen, but achieve other effects instead.

For example, CHAR 7 is a bell. That is, it was originally a bell on older teletypes and other terminals. On most microcomputers, it now sounds more like a beep instead. Try PRINT1 CHAR 7 in the direct execution mode. If this instruction is inserted as the next to last line in the ARROW procedure, it will produce a sound as the arrow hits the backstop.

If a "balloon" target is placed against the backstop, it will "pop" as the arrow pierces it.

```
TO TARGET
   CURSOR   17   10   PRINT1   [O]
CURSOR 0  0
   END
```

To make use of all three procedures, type them in the following order in the direct execution mode of the text screen:

```
BACKSTOP
TARGET
ARROW
```

ERASING A LINE

Children (adults, too!) derive a great deal of pleasure from powerful effects. One sign of a powerful effect is a large result produced by a small effort. This is one reason children like to use numbers such as 1000 when they first begin with the turtle. This causes the turtle to draw for a long time!

The same sort of effect is found in instructions such as REPEAT 1014 [PRINT1 [+]], which fills the screen with +'s. This also makes it possible to demonstrate the difference between PRINT and PRINT1.

First, fill the screen with a character. Then, observe the effects of the following instruction:

```
CURSOR 0 1 PRINT [ERASE LINE]
```

Compare this with the effect of:

```
CURSOR 0 10 PRINT1 [ERASE LINE]
```

If empty brackets are placed after a PRINT statement, nothing is printed. This has the effect of blanking the line. Observe the effect of the following:

```
CURSOR 0 20 PRINT [ ]
```

LOGO ON BROADWAY!

This capability can be used to draw and erase a marquee which marches across the screen, just like the one in Times Square! Tail recursion is used to make the procedure call itself over and over again, so the sign does not move across the screen once and stop, as the arrow did.

```
TO MARQUEE
   CURSOR   0   10   PRINT [SEE OUR
PLAY]
   WAIT 30 ERASE.MESSAGE
   CURSOR   13   10   PRINT [SEE OUR
PLAY]
   WAIT 30 ERASE.MESSAGE
   CURSOR   26   10 PRINT [SEE OUR
PLAY]
   WAIT 30 ERASE.MESSAGE
   MARQUEE
   END

TO ERASE.MESSAGE
   CURSOR 0 10 PRINT []
   END
```

This marquee moves across the screen 13 spaces at a time. A slight enhancement makes it possible to show the message more than once and to vary the number of spaces moved by altering a single number.

```
TO MARQUEE.I
   CLEARTEXT
   MAKE "POSITION 0
   MARQUEE.II


 TO MARQUEE.II
    CURSOR :POSITION 10
    PRINT [SEE OUR PLAY]
    WAIT 10 ERASE.MESSAGE
    MAKE "POSITION :POSITION + 1
    IF     :POSITION     >27    MAKE
"POSITION 0
    MARQUEE.II
    END
```

The program first sends the cursor to position 0 on line 10 and then prints the message SEE OUR PLAY. After waiting for 10 counts, the message is erased. The position is advanced by 1. The procedure repeats after checking to make sure that the message will not be pushed off the end of the screen, the procedure is repeated. Changing the number in the line MAKE "POSITION :POSITION + 1 will alter the distance the sign moves between recursions.

Glen Bull is a professor at the University of Virginia, and has taught LOGO courses at the graduate and undergraduate level.

Tipps for Teachers continued

Such computerrel will not replace Walt Whitman, but may lead to some projects such as shape play with words using PRINT and CURSOR.

```
DOWN               J                HAPPY
E  E              I A                 R
E  A             L   C             FRIENDLY
PEAR            L O O K               T
                   U                  T
                   T               MARYLOU
```

FIRST LAST WORD

The first skill in list processing is getting the computer to print messages which can be controlled, corrected, and changed. Creating word products can be as much fun and may have more utility than turtle graphics.

I recommend that teachers who are working with computers get a word processing program for their own use. The cost is from $70 to $700 with the usual relationship between cost and features.

My one word of bias is against word processors which require changing of "modes" for creating, deleting, inserting, and making capital letters. You will recognize these word processors if you have to hit a key to insert a letter, then a key to exit, then a key to delete, then a key to exit, and so on. All this superfluous key-punching gets in the way of the task.

LOGO sets a good example for screen editing, and experience with it will let you recognize immediately a program which puts barriers in the way of your composing and correcting. When you have used word processing, you will be able to help children use LOGO as an introduction.

Classroom writing with the computer will be an exciting addition to the language arts curriculum. However, one issue needs to be raised, if not resolved. When working with turtle graphics, typing seems to be manageable by the hunt- and-peck method. However, typing with text will not be so easy. Will children need touch typing lessons early or can they be delayed to sixth grade, for example? Are their hunt-and-peck habits going to be as hard to break as those of adults? After you have had some experience with text, let me know what you think.

Steve Tipps is a professor at the University of Virginia, and has presented LOGO workshops for teachers throughout the eastern United States.

Where LOGO Fits continued

Assign each group a quadrant of the mural. Initially, there has to be discussion between groups to decide exactly what will be included in the mural. Then each group separately designs and writes their quadrant procedures.

Once the group has finished, all procedures can be stored on one disk for the final product. This activity will provide positive reinforcement for students needing it along with practice in group problem solving, cooperation, organization, and efficiency. Saving all procedures on one disk may clear up notions of how workspace and disk storage relate.

In language arts we stress the importance of precise spelling and grammar. The same is true in LOGO. Punctuation can also be related to the use of quotes, dots, and square brackets in LOGO. Learning new reading vocabulary can be associated with teaching the computer new words.

There are several class assignments in which planning ahead is very important. When writing reports or stories, students should not simply sit down and write. They must first prepare an outline to help with the organization.

The same type of planning must take place while constructing a LOGO project. Students may not be convinced that it is necessary to write an outline of their report on whales, but a LOGO procedure for a whale consisting of body, flipper, and airhole certainly must be "outlined" first.

There are several ways to do "take offs" on science topics. For example, while studying the solar system, challenge students to write LOGO procedures to draw the planets in their correct order with the proper size relative to one another. While studying food chains, suggest that they develop procedures to draw the animals, fish, and the like in the proper order of the chain. Part of a student science fair project might include LOGO procedures to illustrate a concept.

Of course LOGO and art are a good pair. Use of color, symmetry, asymmetry, and perspective drawing can all be explored.

The subject that obviously has been omitted here is mathematics. I have done this purposely because math is the first subject with which any computer work is associated. Therefore, I hope to encourage teachers to combine other subject areas with the computer and with LOGO.

Bonnie Rifkin is an educational consultant, teacher, and an intern at the Technical Education Research Center (TERC) in Cambridge, MA.

NLX Pen Pals Wanted!

We have two unmatched classes in the NLX Pen Pal Program. If you and your class have interests, grade levels, and experiences which parallel either of those listed below and you would like to be matched up, please let us know right away!

The only suggested requirement is that you agree to correspond with each other at least once a month.

TX-5-TI-1. This is a class of 25 fifth graders in Texas. They have been using TI LOGO since November, and have worked with both the turtle and sprites. They would like to be matched with a school in the southwest, if possible.

AK-78-MIT-2. This class of 15 7th and 8th graders in Alaska has been working with MIT LOGO since September. They are just beginning to write procedures which contain subprocedures. There are also several other students in the computer club who have worked with the robot turtle and 3-D designs. They are eager to correspond with other students who are also working with LOGO.

If you would like to be matched with either of these classes, or with others which become available, please send us the grade level and age of your students, how long they have been working with LOGO, a brief description of their recent LOGO activities, what computer and LOGO version you have, how many students you have, and your name, address, and school telephone number.

PENDOWN

Sandy Towberman sends the adjacent cartoon of "Tommie Turtle," along with an interesting suggestion for creative writing classes.

Why not have children design and caption their own cartoons drawn with LOGO, similar to the one she sent in? Some years ago, it was popular to put captions with doodles. Use this same idea with LOGO drawings. Since Valentine's Day is coming up, how about some funny quotes to go with a LOGO-drawn heart? (One of Sandy's suggestions had Tommie just finishing a heart on the screen and saying "I'm in the mode for love!")

From a creative writing project such as this, it is a small step to put together a booklet which features cartoons by your class. Perhaps the school could "publish" it and have a copy catalogued into the library. (What a time for a lesson on the Dewey Decimal System!)

Sandy Towberman is an artist, writer, and musician who, with her husband, a Lutheran clergyman, is raising a daughter in Canton, Ohio.

The Single Computer: How to Cope continued

The other teachers also gained confidence about using the computer and several of them now plan to use it in their classroom activities. As we get more computers and as more students and grade levels become involved, the scheduling problem will become even more difficult. But that is something we can work out.

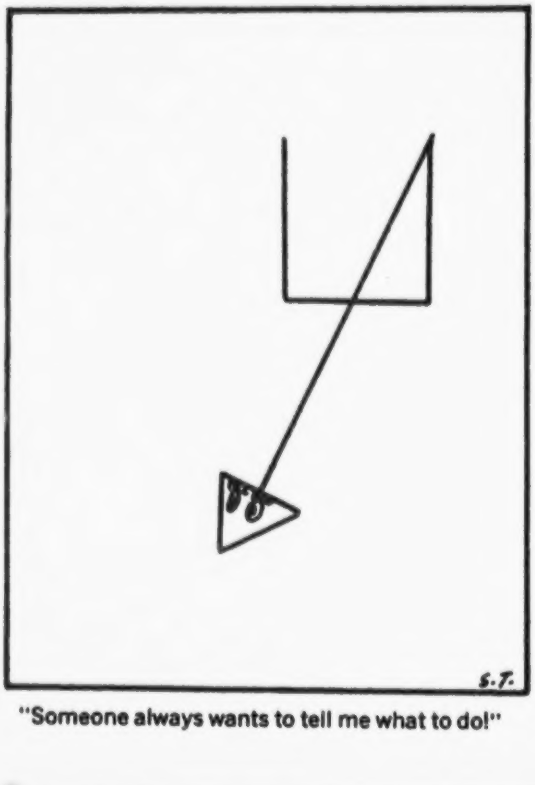
A FINAL NOTE

LOGO has been a great motivator. One boy who could not seem to learn his multplicaiton facts mastered them so he could be allowed time to use the computer, and is now a regular after school "drop in."

The more the students learned, the more they wanted to learn. But, they also learned that programming takes a lot of time and patience. They now appreciate the complexity and time involved to develop the compter games they love to play.

Linda Nix is a fifth grade teacher at Forest North Elementary School in the Round Rock Independent School District, Austin, Texas.

Tommie Turtle by Sandy Toberman



"Someone always wants to tell me what to do!"

The National LOGO Exchange

ISSN 0734-1717

March 1983 FORWARD 100! Volume 1 Number 7

Burnett's **LOGO** Book is for **Do-ers**

LOGO: An Introduction, by J. Dale Burnett, is a special kind of primer, designed to guide a teacher using LOGO in the classroom. It purports to be neither a text nor a manual, but an introduction to LOGO, and its pedagogy and philosophy.

Throughout the 68-page book, one finds questions such as, "How do you explore a procedure?"; "So what?"; "How much structure do we give to student exercises?"; "What should we as teachers do?"; and "Why do we assume such questions have short answers?" Thus, Burnett brings the reader face to face with important LOGO teaching considerations.

Suggested exercises, powerful ideas, and stimulating operations form an interesting mixture. Turtle graphics explorations make up the bulk of the book, with word and list operations being confined to the last 6 pages. This is appropriate for the scope of the work.

Burnett devotes 12 pages to an exploration of symmetry, and shows clearly how this concept could be explored. Most significantly, he shows how teachers can facilitate this exploration. Obviously, he has spent time in the classroom!

At first, I thought the book might be best suited for elementary teachers. I changed my mind. I believe any teacher could be enriched by a study of this book.

Although possessing minor flaws (don't we all??), such as waiting until page 12 to tell the reader that all LOGO commands in the book are for the MIT version for the Apple, it presents LOGO in an educationally sound manner, consistent with the philosophy advocated by Papert, that the child takes charge of his or her learning.

LOGO: An Introduction, by J. Dale Burnett, Creative Computing Press, 39 East Hanover Ave., Morris Plains NJ 07950, (201) 540-0445, $7.95. ISBN: 0-916688-39-9.

HOW TO MAKE YOUR OWN SPRITE GRAPHICS

by

Jim Fry

You can make sprites come alive on your Apple with MIT LOGO, by using Micromint's E-Z COLOR GRAPHICS SYSTEM board and a set of sprite procedures available from Terrapin, Inc.

It all started for me when I read Steve Ciarcia's Circuit Cellar article in the August 1982 (LOGO) issue of Byte magazine. Entitled "High-Resolution Sprite-Oriented Color Graphics," the article discussed sprites and their uses, and showed how to construct a color graphics interface to enable the Apple to use a sprite generator chip.

I contacted Micromint Inc., 917 Midway, Woodmere, NY 11598, (800-645-3479 to order) and made arrangements to purchase their E-Z Color Graphics System board, on which Ciarcia's article was based. It cost $150.00 for the kit and parts, or $175.00 for the assembled board.

A telephone call to Terrapin (617-492-8816) was all I needed to get a copy of their Terrapin LOGO Sprite Procedures disk ($15.00 plus $2.00 postage). They enclosed a demonstration program and a 5-page manual by Leigh Klotz, in which the procedures were explained. It also showed how to enter user defined shapes, since there was no shape editor.

The sprite commands were similar to those of TI LOGO in that they TELL which sprite to CARRY a shape, and set the x and y coordinates and the color.

However, there were no velocity commands to move the sprites, such as the SETSPEED command from TI LOGO. Instead, I had to write procedures which changed the x and y coordinates continuously within a REPEAT statement.

continued on page 2

The National LOGO Exchange, copyright 1983 Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly September through May, $25 per year, mailed FIRST CLASS from Charlottesville, VA. Add $5 per year for addresses outside the USA. Single copy price $3. The opinions expressed by the authors are not necessarily those of The National LOGO Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress St., Salem, MA 01970. Address all other correspondence to The National LOGO Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717/83 $2.00

Editor......................Tom Lough

Get Ready!

For many things, growth often occurs in leaps and bounds, rather than by a steady process.

We see evidence of this all around us. For example, an athlete in a vigorous training program will experience periods of "stagnation," with little or no progress toward his or her goal, interspersed with sudden improvements in performance.

A child's intellectual growth may develop this way also, marked by a step-like pattern of progress. During the "non-growth" periods, it is important for teachers to be patient, knowing that processes which cannot be detected are at work.

The rate of LOGO's growth has been a concern to some. After a period of initial activity with the TI and Apple versions, LOGO seemed to settle down upon a plateau. But all has not been dormant.

Digital Research's recent LOGO announcement has far-reaching implications, for example. The indications are present for a LOGO move toward becoming a general use language. If this proves to be true, we are going to see a healthy spurt of growth! The role of LOGO teachers will become significant, to be sure.

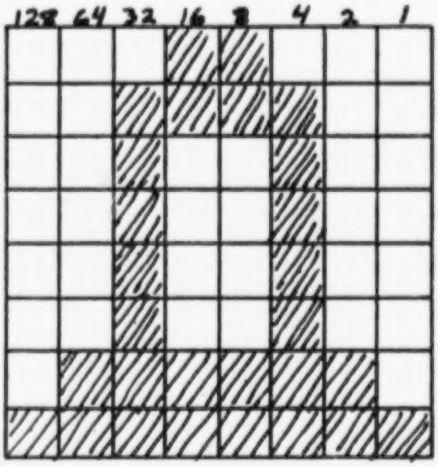
Publications often provide reliable indications of growth. A recent survey of computer books revealed about 10,000 titles in print for BASIC, a dozen or so for Pascal, and only one or two for LOGO! Editors of publishing houses assure me that many LOGO books are in the works. When these become available, we will experience another LOGO step forward.

As LOGO teachers, you are in the right place at the right time! Are you ready for the coming growth? Don't be too smug ... a child is gaining on you while you are reading this!

FORWARD 100! Tom

Sprites Continued

Different shapes can be designed easily and saved on the disk. An 8x8 grid is used, similar to the one below, with numbered columns across the top as shown. The blocks are colored in to show the design, a rocket for example.



In order to tell LOGO the shape, it is necessary to total the column numbers of the blocks in each row. Here, the first row has blocks 16 and 8 colored in, so the total is 24. The second row has blocks 32,16, 8, and 4 colored, making a horizontal total of 60, and so on down the rows. All eight sums are then placed in a list and inserted in the INITNAMES procedure on the Terrapin disk with a command such as the following.

```
MAKE "ROCKET  [ 24 60 36 36 36 36 126
255 ]
```

That's all there is to it! I have found the sprites very enjoyable and exciting to work with. I hope that Terrapin will be offering them as primitives in a newer version of Terrapin LOGO.

If any NLX readers need more information or want to ask any questions, write to Jim Fry, Novi Woods Elementary School, Novi, MI 48050.

(Ed. note: Krell Software Corp., 1320 Stony Brook Road, Stony Brook, NY 11790, offers a complete sprite system ready to go for $325.00. No user modifications are necessary.)

Jim Fry teaches math, reading, and LOGO to elementary school students in Novi, Michigan. He teaches a series of summer computer courses also.

TIPPS for TEACHERS

by

Steve Tipps

Random Thoughts

Introducing PRINT at an early stage in the LOGO classroom is important in overcoming the misconception that LOGO is only turtle graphics. Some interesting projects are possible without reliance on variables or list processing commands. At the same time printing projects can introduce variables and list processing in a natural way.

The INVITATION project from last month's column used a local variable as input which could be changed to personalize the invitation. Children can also write their own arithmetic drills with LOGO and extend problem solving, programming, and procedural skills.

Let me repeat: children should write their own drill procedures. In this way, they learn more about the number system and arithmetic operations by commanding the computer. As the turtle world is a microcosm of geometric rules, working with arithmetic operations in PRINT can help increase their understanding of the operations.

Two new LOGO commands are needed -- RANDOM and SENTENCE. Combined with PRINT, and the MIT LOGO commands of PRINT1 and CURSOR, these commands enable the creation of many new procedures.

A RANDOM PLACE TO START

RANDOM is fundamental to understanding statistics and probability. It expresses the idea that any of a set of events is equally likely to happen. If you flip a coin, either heads or tails will land up. The coin standing on edge is possible, but so unlikely that it is not counted. Therefore, heads and tails are equally likely results of a fair coin flip.

Each coin flip is separate and independent of every other flip. After ten heads in a row, the next flip still has an equal chance of being heads or tails. The common sense law of probability would say that, after 10 heads, the next one should be tails. The common sense law applied to one occurrence is a nonsense law. The coin does not remember or care what has landed before or after. Heads or tails is unpredictable based on prior information. It occurs by chance or randomly.

The same idea applies to the selection of random numbers. If you have a fixed set of numbers (say, 0 to 24), and pick one, any of the 25 numbers has an equal chance of being selected. If you ask the children to guess which one will be drawn, they should be able to pick the correct one only by chance, not with any regularity.

Of course, I've always had students who did not grasp the idea of RANDOM. They took any correct guess by anyone as evidence that the number could be predicted. This is an example of pre-scientific or pre-formal thinking. Teachers can use LOGO to provide experiences with RANDOM. Remember, however, that both maturation and experiences are needed for cognitive development. Dramatizing RANDOM by having the turtle select numbers from a hat may serve as a bridge to understanding RANDOM in LOGO.

Numbered cards from 0 to 9 can be put into a hat or box and drawn out one at at time. Each number has an equal chance of being picked. Be sure to put the card back in the box before choosing again. The child playing turtle can pick, record, and replace. After forty or fifty selections, the tally of each number should be about the same.

The range of numbers can be changed from 0 - 9 to 0 - 24 or 0 - 99. Children would see that, if the range of numbers is bigger, the possibility of guessing what will be drawn is much less. This process is another example of moving from real experiences in the classroom to the abstract level of having the computer perform a function.

RANDOM NUMBERS

Children should be very familiar with the results of PRINT statements such as:

```
PRINT 6
PRINT 6 + 3
PRINT [ 6 + 3 ]
```

The difference between printing the numerals or characters in brackets and the operations and numbers without brackets needs to be well established before introducing RANDOM.

RANDOM can be explained as the turtle pulling a number card from a hat, so that

```
PRINT RANDOM 10
```

means that the turtle has a hat with the first 10 numbers in it: 0,1,2,3,4,5,6,7,8, and 9. The same statement can be put in a REPEAT statement to verify that only those numbers are in the turtle's hat.

```
REPEAT 10 [ PRINT RANDOM 10 ]
```

continued on page 4

Tipps for Teachers continued

Students can keep track of the range and frequency of the RANDOM numbers picked. If your students have had experience with a printer, this could help record the numbers selected also. They need to see that, in the short run, one number can turn up 3 or 4 times, but, in the long run, all the numbers occur about equally.

RANDOM can also be used in turtle commands to have the turtle perform a curious little squiggle closely related to the "random walk" of statistical physics.

```
     FORWARD RANDOM 100
     RIGHT RANDOM 30

     REPEAT  10  [ FORWARD RANDOM 50
RIGHT RANDOM 20 BACK RANDOM 50 ]
```

PENCOLOR and BACKGROUND can also be set with RANDOM as long as the range is compatible (that is, RANDOM 7 will cover the maximum range of 0 to 6, for example). If an incompatible range is set for PC or BG, LOGO will complain eventually that it does not like a certain large number as input!

Another example of RANDOM use is with CURSOR. This command positions the cursor (the blinking box) on the TEXTSCREEN in any column 0 to 39 left to right, and in any row 0 to 23 top to bottom. To print HI! at random locations on the screen, a short procedure could be defined:

```
     TO PRINT.HI!
         REPEAT  10 [CURSOR RANDOM 39
RANDOM 23 PRINT1 [HI! ] ]
         END
```

Sometimes this procedure will "knock the word off" if the first CURSOR position is too far to the right. This can be adjusted by changing the range to RANDOM 36.

A lot of exploration of RANDOM should provide a background for some of the future projects using variables. Besides specifying a number of turtle steps, children should be aware that there are other ways to give that information to the turtle. RANDOM can be a bridge to many uses of variables.

A LITTLE DRILL

At the same time RANDOM is being developed, children can be making simple math drills with short procedures of PRINT statements:

```
     TO PRACTICE.NINE.TIMES
        PRINT [ 9 X 9 = ]
        PRINT [ 9 X 3 = ]
        PRINT [ 9 X 7 = ]
        PRINT [ 9 X 6 = ]
        END
```

Since the concept of interactive procedures using REQUEST and READCHAR are usually introduced at a later time, the children could work these problems on paper.

The format of these procedures can be changed with CURSOR and with PRINT [ ] to provide blank lines and make them more readable. I also suggest adding a WAIT procedure between PRINT lines. Apple LOGO (by LCSI) has WAIT as a primitive. You will have to define WAIT yourself in MIT LOGO. One which we use a lot is as follows:

```
     TO WAIT :TIME
        TEST :TIME < 0
        IFFALSE WAIT :TIME - 1
        END
```

In this WAIT procedure, the larger the number input for the variable TIME, the longer the wait. Thus, WAIT 20 takes longer to execute than WAIT 10. I would be in favor of giving children WAIT as a "primitive" and not belabor its construction. It is a handy idea that facilitates many procedures. A sample procedure for a timed math drill with answers follows:

```
     TO PRACTICE.NINE.TIMES.W.ANSWERS
        NODRAW      ; CLEARS TEXTSCREEN
        CURSOR 5 5 PRINT1 [ 9 X 9 = ]
        WAIT 125  ; WAIT FOR ANSWER
        CURSOR 20 5 PRINT 9 * 9
        WAIT 75     ; NEXT PROBLEM
        CURSOR 5 8 PRINT1 [ 9 X 3 = ]
        WAIT 125
        CURSOR 20 8 PRINT 9 * 3
        WAIT 75
        CURSOR 5 11 PRINT1 [ 9 X 7= ]
        ...
        ...
        END
```

The children could try to write their answers on paper before LOGO prints the correct answer on the screen.

Children can move from PRACTICE.NINE.TIMES to this procedure with very little guidance if they understand PRINT1 and CURSOR. They should be able to speed up and slow down the display by changing WAIT. But here, each problem has to be written separately. A more "economical" technique can be used if RANDOM is fully understood.

A LITTLE RANDOM DRILL

RANDOM makes it possible to produce many different problems with a simple single line statement such as:

```
PRINT1 [ 'WHAT IS 8 MINUS   ' ] PRINT
RANDOM 9
```

The apostrophe's are needed in MIT LOGO to preserve the spaces in the brackets. This REPEAT statement can be defined as a single procedure and then used in a REPEAT statement.

continued on page 8

MICROWORLDS

by
Glen Bull

Glen Bull will resume his regular column next month. He and his wife Gina are celebrating the arrival of their first child, Stephen Wesley!

Creating a LOGO Environment

by Tim Riordon

(Editor's note: We believe this LOGO article to be one of the most significant written to date. We are pleased to be able to share it with you. It is presented here in serial form. Copyright 1982 ICCE. Reprinted with the kind permission of The Computing Teacher and the author.)

(Author's note: Perhaps you are a curriculum specialist, school principal or classroom teacher interested in introducing LOGO in your district, school or classroom. You've done some reading and have a grasp of the philosophy. Now, however, it's time to implement your ideas. How do you actually create a LOGO environment? This article bridges the gap between theory and the beginning of actual practice.)

The training in LOGO that many teachers are receiving focuses upon learning the LOGO language. In four hours, a day, or perhaps a week, teachers begin learning to use LOGO commands to write procedures to do wonderful things - graphic designs, music, robot control, random poetry, arcade-like games, etc. This training is necessary but not sufficient for most teachers to use LOGO as it was intended. Most teachers need not only to learn the language but also to learn how to implement LOGO - how to create a LOGO environment in the classroom.

This article examines five questions:

* What is a LOGO environment?
* What are its attributes?
* What things are conspicuously absent in a LOGO environment?
* What are the reasons for creating such an environment?
* How does one begin?

Ideally, a LOGO environment would have a number of computers with many students doing LOGO at the same time. However, such environments will not occur in most schools until school districts begin to pool computer resources and lend individual teachers sets of LOGO machines. Therefore, this article is written for the classroom teacher who has only one computer and wishes to create a LOGO environment.

WHAT IS A LOGO ENVIRONMENT?

It is the entire context, made possible and managed by the teacher, in which students work with LOGO. It is more than a computer learning station. It includes psychological as well as physical space - how students feel, how students and adults interact.

WHAT ARE THE ATTRIBUTES OF A LOGO ENVIRONMENT?

What would an observer see, hear and feel? Suppose an omnipresent observer - perhaps a principal doing an evaluation - were to spend a week in a classroom with 25 students, one computer, and a LOGO environment. What would s/he observe?

It is Monday morning. Teacher and students have not yet arrived. In the front corner of the room, not far from the teacher's desk, is a table with two chairs, a computer with disk drive, a large television monitor and supplies. Supplies include masking tape, rulers, graph paper, pencils, felt tip pens and drawing paper.

There is a box with three disks in it labeled "LOGO STORAGE DISKS." A small weekly schedule shows each day divided into 20-minute periods. The schedule is nearly filled with students' names. The words "Whole Class" appear four times on the weekly schedule. Each student uses the computer 3 times per week while the rest of the class continues with other instructional activities.

On one wall is a chart with student names down one side and several projects listed across the top of the chart. Many students have checked off projects after their names. Above the chart is the sign "TRY TWO OF THESE PROJECTS THIS WEEK," a direction which seems to invite students without commanding or threatening.

On the other wall are some computer printouts of student designs signed by the student programmers. There are also drawings done in felt tip pen - some by students, some by the teacher. These drawings are on a section of the wall labeled "CAN YOU MAKE THE COMPUTER DO THIS?"

This is a problem-solving environment. There appears to be a sharing of authority since both students and the teacher are allowed to contribute to the collection of problems. On the floor are many designs made with masking tape. What purpose do these serve?

The teacher arrives and removes some of the drawings from the walls. Four new problems are posted. Three of them are designs and one is a question: "Can you write a LOGO procedure that will put your name into a code?" The students are evidently using LOGO for more than graphics exploration.

continued on page 6

LOGO Environment continued

School begins and students, sometimes alone and sometimes with a partner, go to the computer at their assigned times. For the most part, transitions are smooth. Kids are eager to get to the computer, reluctant to quit and store their unfinished business.

Once the teacher has to intervene: "Mary, you really must finish now. It's Kevin's turn. There's some open time at lunch recess I'll let you use if you wish."

During the time each student works at the computer the teacher finds a few minutes to watch and sometime, seeing a student encounter an unexpected result, asks quietly, "Do you have a theory about what caused that?" Most interaction with students about their LOGO work, however, occurs at another time.

Occasionally, the teacher will direct something to the entire class, including the student working at the computer. "Kevin, I need to interrupt you so you can hear an announcement for the whole class ..."

Several times throughout the day students are seen leaving the computer, going to another student in the class, interacting and then returning to the computer. Sometimes both students will go to the computer for a short time. Then the student who was doing seatwork will return to his/her seat and continue working.

Listening to these interactions, one hears: "I can't remember how to make the cursor go down." "How do you make the turtle appear randomly at different plaes on the screen?" "SETXY isn't working right." "How do you erase a line you don't want?"

The explanation for this is that each week two students volunteer to be LOGO EXPERTS. Any student experiencing difficulty while working at the comptuer has permission to seek the help of one of these experts. There is one boy and one girl, and it should be noted that the experts are not always the most academically advanced students.

The teacher is sharing authority and at the same time greatly reducing the number of interruptions which would otherwise occur.

About twenty minutes before lunch recess, the teacher ends a language arts lesson and walks over to the computer. The schedule says "Whole Class." A few students sitting in the back move forward in order to see better.

After asking for volunteers to demonstrate some of their work, two students show a procedure that draws a submarine in water. An airplane is then drawn in the sky. A bomb appears beneath the plane and slowly falls toward the sub, causing an explosion when it hits. This demonstration is followed by a student who offers a procedure that draws Garfield the Cat at random places on the screen.

Whenever a student demonstrates a procedure, there follows a discussion, shaped by a questioning strategy composed of four questions:

$(1) Do any class members have questions about the procedure we've watched?

Several students usually want to know how it is done, so the demonstrators show the inner workings of their procedures.

Occasionally, a value-related question arises when a student refuses to show how a procedure works. "I don't want to show it - it's a secret." This leads to a brief discussion about who owns knowledge, and about copyrights and patents. (Usually this motivates one or two other students to discover the secret which is later gleefully published!)

The second and third questions are addressed to the student who demonstrated the program:

$(2) What do you plan to do with your procedure now?

Sometimes the answer is, "Nothing - I'm done with it." Other times elaborate plans are given. Frequently, the student will respond in a sketchy, tentative manner as in these examples: "We'd like to have it so that a person can aim and fire the bomb." "I'd like to have some music go with the drawing." "I'm going to try to figure out how to make each cat a different color."

The tentative nature of these responses tells the teacher that the student has encountered a new problem to be solved but does not yet know the solution. Such responses are always followed up by another question:

$(3) Do you have some ideas to try out or do you need some hints about how to do it?

The teacher is seen making notes for students wanting help. The notes read like this:

Ben - give him music routine
Sarah - have Ann show SETXY and RANDOM
Thane - hint about multiple colors
Keri - tape and walk rainbow starting from center
Raol - no help needed

Several things about this environment are becoming evident. Students decide for themselves whether they need help solving a problem and in so doing they learn to reflect upon, and be responsible for their own learning.

continued on page 8

LOGO in the News

What a month for LOGO! Many new and exciting things are happening! New versions are on the way, lower prices are announced, and a robot turtle from Australia is now making tracks in the USA.

NEW LOGO VERSIONS

Digital Research has recently announced the development of DR LOGO. (trademark Digital Research). It is reported to contain all of the features currently available on the popular LOGO versions, and some very welcome additions. For example, system information and command explanations can be called up on the screen, eliminating searches through reference manuals. Other features include the ability to use indentation, string processing, and several debugging aids.

Comment statements are allowed, but DR LOGO also provides a method to erase all comments, if desired, once a procedure is debugged. The text screen and the editor may be viewed simultaneously. Upper and lower case letters may be used.

From the press releases, one gets the idea that this will be a very powerful version of LOGO, targeted at both the "traditional" LOGO market and the business and professional community. In his article in the March 1982 issue of Microcomputing magazine, Harold Nelson suggested that LOGO had the potential to surpass BASIC as the most widely used general purpose computer language. Gary Kildall, Digital president, supports this by declaring that "LOGO promises to become this decade's most widely accepted computer programming language."

If anyone should know, it would be Kildall. He is the author of the phenomenally successful CP/M system.

DR LOGO will be available initially for the IBM PC and the TI Professional Computer. LOGO on the IBM is exciting to contemplate. With the present versions (Apple, etc.), at best only a few thousand nodes of memory are available for use. DR LOGO will offer nearly 100,000 nodes!!! No more "OUT OF MEMORY" messages!

The only modification necessary for the IBM PC is the installation of a color graphics card.

These versions may be available as early as Fall of 1983. DR LOGO may be the most compelling reason ever to buy a computer!

EVEN MORE VERSIONS

Atari has announced a contract with LCSI to develop LOGO for their 400, 800, and 1200 computers. Atari's LOGO will be a plug-in cartridge with full features. With a price under $100 and a hoped-for availability of this summer, This LOGO version is going to make quite an impact. If your school is one of the thousands with Atari computers, keep your eye out for Atari LOGO!

Michael Burke of San Jose State University (California) reports that LOGO for Mattel's Aquarius computer is being developed, and may be available within the next few months. It will include a full-feature language, with multiple turtles and animation functions. Two types of turtles, toilers (line-drawing) and tilers (graphical shape drawing), will provide many new capabilities. This may be the most powerful inexpensive LOGO version available.

STILL MORE VERSIONS

Available now from Cybertronics International, Inc. is their CyberLOGO for the Apple II 48K. Costing $99.95, this version features turtle graphics, sound, a full screen editor, and the ability to save both procedures and pictures.

A feature I particularly like is its ability to ferret out LOGO primitive commands from keyboard input. For example, if you were to enter WOULD YOU MARCH FORWARD ABOUT 70 STEPS PLEASE, the CyberLOGO turtle executes FD 70. This seems to appeal to younger children, who want the computer to be their friend.

Although limited to integer arithmetic and interrupted by infrequent disk accesses, CyberLOGO is worth investigating, especially if all you really need is turtle graphics.

Don't overlook Radio Shack's Color LOGO! It, too, has been available for some months. A disk version for the TRS-80 Color Computer costs $99.00, and runs on a 32K machine. A cartridge for under $50.00 for the 16K machine will be available soon. The most talked about feature of Color LOGO is its HATCH command, creating up to 255 turtles on the screen, each of which can draw!

KRELL LOGO PRICE SLASHED

As of March 1, Krell LOGO will cost $89.95! Including 4 disks (2 MIT LOGO disks, 1 demo disk, and 1 Krell utility disk), a technical manual, LOGO wall chart, and a premier copy of Krell's LOGO and Educational Computing Journal, this promises to be the best price break around for Apple owners. Krell has also made available a $325.00 sprite board for the Apple II.

continued on page 8

Tipps continued

```
     TO EIGHT.MINUS
        PRINT1 [ 'WHAT IS 8 MINUS  ']
PRINT RANDOM 9
        PRINT [ ]
        END
```

```
     REPEAT 10 [ EIGHT.MINUS ]
```

This will make ten problems, most of which will be different. However, the same problem can also be written with one PRINT statement, if SENTENCE is used. SENTENCE is a LOGO comand which connects two or more elements.

```
PRINT ( SENTENCE [ WHAT IS  8 MINUS ]
RANDOM 9 [ ? ] )
```

This is a number sentence which has three parts: WHAT IS 8 MINUS, the RANDOM number to be subtracted, and a question mark. SENTENCE connects the three parts into a sentence. Note that no apostrophe was used.

Students can generate different combinations of number drills by varying the operations and the size of numbers.

```
     TO RANDOM.ADD
        PRINT  ( SENTENCE RANDOM 1342
[ + ] RANDOM 888 [ = ] )
        END
```

Many different drill programs can be written by children for their own entertainment and learning, or as puzzles for their friends. These RANDOM procedures can be incorporated into more elaborate drill programs, such as in the previous section.

A LITTLE LESS THAN RANDOM

After the children have finished their programs and run them several times, they may notice the same sequence of numbers turning up in the problems. This becomes especially obvious if they have only one math procedure.

The way RANDOM works is not as fair as the turtle pulling numbers from the hat. Actually, RANDOM is a list of numbers which the turtle reads. The turtle starts at the top of the list, unless it is instructed to go somewhere else and begin reading.

The command necessary to start the turtle at a different place in the list is RANDOMIZE. In the early stages of the children's work, this command may not be needed. But, when children complain that the turtle always picks the same numbers, you should then introduce RANDOMIZE as the first line in any procedure which uses RANDOM.

Steve Tipps is a professor at the University of Virginia, and has presented LOGO workshops for teachers throughout the eastern United States.

LOGO Environment continued

Some students who want help are given the complete answer to their problem, sometimes by the teacher and sometimes by another student - authority is distributed. Students often are given hints but not complete answers; when guided discovery is used, students frequently experience the joy of finding the solution by themselves.

The final question raised after each demonstration leads to a short brainstorming session by the class:

#(4) What are some things which could be done to change this program?

Many fascinating LOGO projects result from ideas contributed from several class members. The teacher also contributes ideas - although usually after ideas have emerged from students.

The whole class responds to work demonstrated by class members two other times during the week. And on one other occasion, the teacher demonstrates a new LOGO programming idea to the entire class. (Earlier in the year when LOGO was first being introduced, teacher demonstrations outnumbered student demonstrations. Now the balance has shifted.)

To be continued next issue.

Tim Riordon is a teacher and consultant in Eugene, Oregon, and is a regular contributor to The Computing Teacher magazine.

LOGO in the News continued

TURTLE TRACKS FROM DOWN UNDER

Harvard Associates, Inc., have announced the Tasman Turtle, an easy-to-use programmable robot turtle manufactured by Flexible Systems of Hobart, Australia. The Tasman can operate from an Apple II or other parallel interface computer, as well as from an RS-232 interface. It can move, turn, toot a horn, blink "eyes," draw with a pen, and "feel" with touch sensors. Extra options include a voice capability and an electronic compass. The price is $999.95.

TERRAPIN ANNOUNCES

Terrapin LOGO will run on both the AROS network by SWI and the Corvus Omninet, according to a recent announcement. Further, Terrapin LOGO is reported to run on both the Franklin and the new Apple IIe computers if the commands are typed in upper case letters.

The National LOGO Exchange

ISSN 0734-1717

Volume 1 Number 8 FORWARD 100! April 1983

Tessalate Your NLX ABC's

by Jim Fry

After receiving NLX's great set of procedures for drawing the alphabet, I immediately typed them in, debugged, and saved. I then proceeded to try them out. "WOW!" is all I can say! They worked fantastically! My community education advanced LOGO class was going to be starting in two weeks and I knew the kids would love using the alphabet.

I saved the alphabet procedures on the 14 disks I use for my class. (I am very lucky to have a computer lab with 14 Apple II's in one room at our high school for my LOGO classes.) When the students arrived, I introduced them to the ABC's and off they went!

Of course, the first thing everyone did was to spell out his or her name. Then, some of them started to have the turtle repeat the same letter over again to make a pattern.

Just the week before, I had been reading the School House Apple article on tessalations. (SOFTALK January 1983, page 228, by Jim Muller of the Young Peoples' LOGO Association.) I had been trying to think of a way to do tessellatons with my students. Of course! Let's use the NLX Alphabet!

I talked to my students about patterns and repeating the same shape or set of shapes over and over again. They had already started doing tessellations with the alphabet while they were playing with the letters. We decided to set up a procedure called PATTERN.

```
TO PATTERN
  PU SETX -125
  FD 70 PD
  REPEAT 20 [PD PC 2 M PD PC 4 K]
  END
```

This was our first procedure. The students picked the letters and colors they wanted. We used all the PENDOWNs because a lot of the letters leave the pen up after they are drawn. (Ed. note: This shortcoming has been corrected in the newly revised version now available.) The students also experimented with different size letters and different amounts of repeating.

continued on page 2

Ed Emberley and the Turtle

by Sharon Burrowes
assisted by David Burrowes

This year, I have found myself rather suddenly involved in helping teachers find ways of using LOGO in their classrooms. I am always on the lookout for new ideas to suggest from whatever source I can find.

Last year, as I was just beginning to experiment with LOGO procedures, my husband remarked that the ability to combine predefined shapes reminded him of some drawing books our children had used when they were small. I agreed, filed it under "things to think about someday," and went back to trying to learn LOGO!

Then, just recently, I received a copy of the NLX ABC's. After keying them in and having a delightful time fiddling with them, I remembered my husband's remark of last year, and set off to scrounge for the drawing books.

To my delight, I found them exactly where I remembered them! The books, ED EMBERLEY'S DRAWING BOOK OF ANIMALS and ED EMBERLEY'S DRAWING BOOK, MAKE A WORLD (Little, Brown Co. Boston), each suggest a "vocabulary" of letters, numbers, shapes, and "things" (such as curlicues) that can be used to create any of the delightful drawings in the book.

Since I already had the NLX letters, I set out to complete the "vocabulary" necessary for the DRAWING BOOK OF ANIMALS. I made some modifications to the NLX letters, and added most of the numbers, shapes, and "things" that Mr. Emberley suggests. Before I had finished, my high school aged son, David (who loved the books when he was younger), got involved in the project and began creating procedures to do some of the drawings in the book.

I heard many mutterings of "Almost!" and "Darn!" over the next several hours. Later, he commented that doing these drawings was not easy ... and I would agree.

The books are aimed at free hand drawing, but they are definitely procedural in concept. Judicious choices of drawings from the books could result in many challenging LOGO project ideas on which children could build.

continued on page 2

The National LOGO Exchange, copyright 1983 Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly September through May, $25 per year, mailed FIRST CLASS from Charlottesville, VA. Add $5 per year for addresses outside the USA. Single copy price $3. The opinions expressed by the authors are not necessarily those of The National LOGO Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress St., Salem, MA 01970. Address all other correspondence to The National LOGO Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717/83 $2.00

Editor......................Tom Lough

The Turtle Elite

During the Middle Ages, one of the sources of power for the Church was literacy. The ability to read and write was guarded carefully, and was transmitted only to selected persons. Such was the power of the written word.

With the arrival of computers, the potential exists for a similar elite class to arise. If computer literate educators are not willing to share with others, there is cause for concern.

Let's face it. The computer using teacher will be able to accomplish much more than a "regular" teacher. Using a word processor to create handouts and other lesson materials can save countless hours, for example.

Similarly, a LOGO teacher will be able to create within his/her classes an excitement difficult to match with traditional methods. Children will be learning in new ways with a powerful tool of today's electronic age. Thus, access to the computer and knowledge of its capabilities become key points of concern.

I can imagine nothing more stultifying than a clique of LOGO teachers who dare anyone to join them, or to try to arrange for access to "their" computers. As we learn more and more about this incredible language and its potential for learning, let us all resolve to reject the idea of a LOGO elite.

If we are not willing to share our knowledge and facilities with others, the entire profession suffers. Even more importantly, the children suffer.

OUTWARD 100!

Tom

Emberley continued

So, if you are looking for some ideas to suggest to your children using LOGO, why not obtain Ed Emberley's wonderful drawing books and a copy of the procedures for the NLX ABC's, and try some experimentation yourself?

(Ed. note: To get your copy of the NLX ABC's, send a stamped self-addressed envelope to NEW NLX ABC's, PO Box 5341, Charlottesville, VA 22905. Several of the "thing" procedures referenced above are included with the listing. Also see the April 1983 issue of The Computing Teacher.)

Sharon Burrowes is a consultant and a computer science teacher in the Wooster (Ohio) City School system.

Tessalate continued

They then started having the turtle draw the PATTERN procedure all over the screen by typing PATTERN many times. We then made a procedure called DESIGN.

```
TO DESIGN
   REPEAT 10 [ PATTERN ]
   END
```

By playing with the REPEAT input number, the students were able to fill up the entire screen with some very interesting designs.

The students found out very quickly that letters with any type of curve took a long time to draw, so they used only letters like A, E, H, K, M, etc.

I have been working with the alphabet tesselations and have come up with some other procedures that I plan to show the students and let them experiment with.

My students have had a great deal of fun with the alphabet and I thank Tom Lough and the NLX for putting them together and sharing them with all of us LOGO fans.

FD 100! Jim Fry

Jim Fry teaches math, reading, and LOGO to elementary school students in Novi, Michigan. He teaches a series of summer computer courses also.

EdCom Postponed

EdCom is postponed! According to its organizers, the EdCom Conference scheduled for late April in Washington, DC, has been rescheduled for October 15, due to a "conflict with another major conference."

TIPPS for TEACHERS

by

Steve Tipps

Variables: More than one way to tame a turtle

Achieving power with a computer is letting the computer do things which relieve humans of drudgery and allow creativity. Repetition is one powerful idea in computing which repeats and recurs. LOGO's meaningful names for procedures, variables, and files are another example of using the computer's abilities rather than humans' to remember things like A1 or X4 or N$.

Variability is another way that the computer reduces work. Actually, variability has been a part of a student's LOGO experience since the first time a decision was made about how far to move or to turn the turtle. The words CHANGE, VARY, and INPUT should be commonplace. Typing PENCOLOR without any numbers reminds the user that PENCOLOR NEEDS MORE INPUTS. (It needs a number to determine which color to use when drawing.) When children have explored and can use economical REPEAT statements, create and embed procedures, and save their work on file disks, they should be ready for making procedures with variable inputs.

ONE WAY AND MORE

Different sizes of the same shape are often needed to construct a project in LOGO. A house may need squares or rectangles for the frame, doors, and windows. A cylinder can be made of circles and circleparts. One modular way is to describe each part of the design in a specific procedure that is unique to that project. HOUSE is composed of FRAME, DOOR, WINDOW, and TRIANGLE. All of the procedures are sized for only that house.

Basic shapes can also be labeled in some relative way. Procedures can be written and named BIG.SQUARE, MEDIUM.SQUARE, LITTLE.SQUARE, or GIANT.CIRCLE, MEDIUM.HALF.CIRCLE, ITSY.CIRCLE.PART. These names are partially descriptive; however, they will proliferate and are not fully descriptive. What is the relationship between BIG.SQUARE and BIG.TRIANGLE, for example?

Children should be able to see that a square can be changed in size by editing the input number for FORWARD. The size of the square is determined solely by the FORWARD input.

```
TO BIG.SQUARE
REPEAT 4 [ FORWARD 99 RIGHT 90 ]
END

TO MEDIUM.SQUARE
REPEAT 4 [ FORWARD 47 RIGHT 90 ]
END

TO SMALL.SQUARE
REPEAT 4 [ FORWARD 13 RIGHT 90 ]
END
```

Comparison of the procedures makes it plain that the size is related to the FORWARD number.

I have suggested to students that they rename their procedures so that they can tell LOGO directly which square they want, rather than trying to remember how big a BIG.SQUARE is. The method I suggest is that they rename BIG.SQUARE as SQUARE.99, MEDIUM.SQUARE as SQUARE.47, and the like.

```
TO SQUARE.99
REPEAT 4 [ FORWARD 99 RIGHT 90 ]
END
```

From this point, children can use a family of squares, circles, triangles, or hexagons to create nested, adjacent or radiating master procedures.

```
TO NEST.TRIANGLES
TRIANGLE.10
TRIANGLE.20
TRIANGLE.30
TRIANGLE.40
END

TO FAN.TRIANGLES
TRIANGLE.10 RT 10
TRIANGLE.20 RT 10
TRIANGLE.30 RT 10
TRIANGLE.40 RT 10
END
```

Projects using procedures named with an indication of the size demonstrates good understanding of the concepts. Although the issue of descriptiveness is solved, two problems remain.

a. The names get long. Children soon solve this by changing SQUARE.100 to SQ.100.

b. Many square procedures are being developed which differ only in one respect: the input number for FORWARD. This diverts the user from good economical programming.

Moving to names with numbers in them serves as a bridge in syntax and provides a lot of experience with variability of a procedure. LOGO teachers will recognize that this naming system is only a stepping stone toward the introduction of variable names and inputs.

continued on page 4

Tipps for Teachers continued

CHANGING WAYS

One or more of the students may be used to asking with regularity, "Isn't there an easier way to do this?" The answer with computers is usually YES. Finding the right time to move on to that easier way is the teacher's dilemma with LOGO. Waiting until children have enough experience to understand the concept is important before introducing the method of the easier way.

SQUARE.99, SQUARE.28, and SQUARE.47 are procedures which differ in size, stated in the procedure name as 99, 28, and 47. By now, children may be able to recognize that all of the SQUARE.xx procedures can be reduced to a single procedure by substituting a variable rather than a number as input to FORWARD.

```
    TO SQUARE :SIZE
    REPEAT 4  [ FORWARD :SIZE RIGHT
90 ]
    END
```

Any meaningful name may be used to refer to the changing value. Typical choices are :SIZE, :LENGTH, or :SIDE. The colon, called "dots" in LOGO lingo, is a signal that a value is expected in order for the procedure to work.

The :SIZE which you add in the naming line of the procedure allows you to input that value directly to the procedure. Adding :SIZE to the name line of the procedure is like a contract or promise to the procedure. Every time you call the procedure, you must give it a value to use for the variable.

Review what happens when you type FORWARD without a number.

```
     FORWARD
     FORWARD NEEDS MORE INPUTS (or
some such message)
     FORWARD 10
```

With the last entry, a familiar result is obtained: the turtle makes a line 10 turtle steps long. Now try to type SQUARE without a number, having defined SQUARE :SIZE as above.

```
     SQUARE
     SQUARE NEEDS MORE INPUTS
     SQUARE 10
     SQUARE 20
     SQUARE 52
```

With the last entries, you have drawn squares of size 10, 20, and 52, all with a single procedure which has had a variable substituted for a specific number.

Two major ideas are explored in this seemingly simple way. First, the idea of variability and establishing a variable microworld is heady stuff. Second, the abstraction of signifiers being used for value is Basic to logical thought.

Coming to grips with variability and signifiers in the turtle's world should prepare children to work with ideas in algebra, geometry, social studies, literature, etc. You may say that algebra and geometry are obvious, but why social studies and literature? I ask you, aren't metaphors and personification, economic theories and cultural rituals rich examples of variability and abstact signifiers?

MANY WAYS

The use of variable inputs to variable procedures may soon generalize to the substitution of variable names for <u>all</u> numbers. One seventh grade class I worked with insisted on changing all the numbers in SQUARE:

```
     TO   SQUARE     :SIZE     :TURN
:NUMBER.OF.TIMES
     REPEAT :NUMBER.OF.TIMES [FORWARD
:SIZE RIGHT :TURN]
     END
```

They discovered that the procedure worked only if three numbers were provided as input. But, they also learned that, suddenly, the procedure name SQUARE was no longer very descriptive!

When challenged to figure out how to make the turtle return to its starting place, they rediscovered the Total Turtle Trip Theorem as a special relationship between :TURN and :NUMBER.OF.TIMES. A new generalized procedure was made up to account for these discoveries.

```
     TO ANY.SHAPE :NUMBER :SIZE
     REPEAT :NUMBER  [ FORWARD :SIZE
RIGHT 360/:NUMBER ]
     END
```

In the new variable microworld, children also found that there was a relationship between triangles, squares, hexagons, octagons, twelve-agons, ... and circles, based on varying :NUMBER.

Working with variables is important, one of the most important steps in programming. Using variables in procedures is a good start, but it is not the end.

Using assignment statements for global variables, initialization of variables, passing parameters between procedures, reading values for variables from the keyboard and paddles in interactive procedures -- all these provide many ways to explore and extend variability. Almost all of the interesting things with list processing in LOGO demand a good grasp of variables. When you help your children discover many ways to tame the variable turtle, you have given them much power in thinking and programming.

Steve Tipps is a professor at the University of Virginia, and has presented LOGO workshops for teachers throughout the eastern United States.

MICROWORLDS

by
Glen Bull

Teaching Grammar to a Computer

One of the best ways to learn is to teach someone else. The foundation of LOGO is based on the child teaching the computer, rather than use of the computer to program the child. Rules for formation of plurals provide an illustration of one way this might be done.

AN ENGLISH DESCRIPTION

The first step in teaching a computer to form plurals involves a thinking of a description of how plurals are formed. A child's first attempt might be:

To form a plural, ADD S TO THE END OF THE WORD.

This English description is the most important part of the process of teaching the computer to form plurals.

TRANSLATING THE DESCRIPTION INTO LOGO

The next question is, "How can this be translated into a LOGO procedure?" As it happens, this is not a difficult problem. The key to learning focuses on the initial description, not minor details of writing the short LOGO procedure. The LOGO instruction WORD can be used to combine words:

PRINT WORD "FIRE "TRUCK results in FIRETRUCK.

The same process can be employed to add S to the end of any item:

```
TO PLURAL :ITEM
OUTPUT WORD :ITEM "S
END
```

This procedure states, "Return as output a word consisting of an item plus the letter S attached to the end." For example,

```
PRINT PLURAL "BOY
BOYS
```

In this instance, LOGO provides a means of verifying the rule. Inevitably, another child will use this procedure in the following way:

```
PRINT PLURAL "BOX
BOXS
```

DEBUGGING THE RULE

The most important learning occurs because the procedure does not work, rather than because it does. In this case, the old rule of adding S must be modified to account for other possibilities. This leads to the discovery that, in general, if a word ends in X, the plural is formed by adding ES rather than S. This rule may be expressed in English in the following way:

To form a plural, IF THE LAST LETTER IN A WORD IS X, ADD ES; OTHERWISE, ADD S TO THE END OF THE WORD.

The last letter in a word can be examined using the LOGO command LAST.

```
PRINT LAST "BOY
Y
PRINT LAST "BOX
X
```

This instruction provides a means of determining whether a word ends in X.

```
TO PLURAL :ITEM
TEST LAST :ITEM = "X
IFTRUE OUTPUT WORD :ITEM "ES
OUTPUT WORD :ITEM "S
END
```

In this modification of the procedure, the last element of the item (LAST :ITEM) would be Y in the instance of BOY and X in the case of BOX. In the first case, the test would fail to obtain a match, and the plural would be formed by adding S. In the second case, the test would produce a TRUE result, and the plural would be formed by adding ES. (Note that the procedure is terminated when <u>either</u> of the OUTPUT commands is executed.)

Again, it must be noted that the mechanics of how the rule is stated in LOGO are not as important as the discovery of the rule itself. LOGO is only being used as a <u>tool</u> to verify the hypothesis.

The role of the teacher is not to write a program which forms plurals for the child. Instead, it is to guide the process into productive areas and away from sterile dead ends, and to ensure that irrelevant mechanical details do not interfere with the learning process.

By this time, there may be some competition among the students to determine who can produce the smartest spelling program. Some teachers may have students compete in teams of two or three, depending on the teaching environment and philosophy.

Microworlds continued

At this point, energies will be equally divided between efforts to improve proprietary PLURAL programs, and efforts to discover words which will fool competing programs. Efforts to discover weaknesses in other programs are just as valuable as improvement of a team's own program, since those efforts also lead to extensions of the rule system. For example, a team might discover that PRINT PLURAL "GRASS produces GRASSS! This presents a problem which could be relieved in the following way:

```
TO PLURAL :ITEM
TEST LAST :ITEM = "X
IFTRUE OUTPUT WORD :ITEM "ES
TEST LAST :ITEM = "S
IFTRUE OUTPUT WORD :ITEM "ES
OUTPUT WORD :ITEM "S
END
```

DETERMINATION OF GROUP MEMBERSHIP

When there are several items to be tested, the command MEMBER? (MIT) or MEMBERP (LCSI) can be used to evaluate the possibilities. This command is used to determine whether an item is an element of a list. This instruction could be used to test whether the last letter of the word were one of S, X, or Z. For example,

```
TEST MEMBER? LAST "BOX [S X Z]
```

would produce a result of TRUE because the last letter X is a member of the list (S X Z). This instruction is a primitive (built-in) command in some versions of LOGO. For those in which it is not (TI LOGO, for example), it is possible to construct this command.

```
     TO MEMBER? :ITEM :LIST
     IF :LIST = [ ] OUTPUT "FALSE
     IF :ITEM  =  FIRST  :LIST OUTPUT
"TRUE
     OUTPUT  MEMBER?  :ITEM  BUTFIRST
:LIST
     END
```

It is not necessary to follow the strategy employed in this procedure in order to use it. The first element in the list is checked to determine if it matches the item in question. If it does, the procedure returns a value of "TRUE. If not, the first element of the list is removed and the process is repeated until a match is obtained or the list is empty. If all the items have been checked without obtaining a match, the procedure returns a value of FALSE.

MEMBER? provides an economical means of testing several possible conditions which would require the addition of ES rather than the S to form the plural.

```
TO PLURAL :ITEM
TEST MEMBER? LAST :ITEM [S X Z]
IFTRUE OUTPUT WORD :ITEM "ES
OUTPUT WORD :ITEM "S
END
```

This version of PLURAL does not encompass all possible methods of forming plurals. There are the cases of suffixes such as CH and SH and of words such as BABY (Readers of last month's NLX will recognize the significance of this word!). The particular methods used to express the rules for plural formation in LOGO are less important than the student's attempts to discover those rules. LOGO simply provides a dynamic means of testing those rules.

CLASSROOM USE

Most of the students who might attempt to describe the rules for plural formation will have had exposure to these rules in previous grades. There is a discrepancy between the level of the content material learned (formation of plurals) and the level of understanding of LOGO required to teach this content to the computer.

LOGO provides a means of exploring these spelling rules in greater depth, even though they may have been learned in earlier grades. In school, the way a rule works is often learned well before the need for a rule is understood. The strange appearance of the word GRASSS on the screen and the difficulty in pronouncing such a word create for the student a reason to find an alternative to the rule ADD S TO THE END OF THE WORD.

In this case, the knowledge of the particular rules may be less important than the sudden insight into why the rules operate in the manner in which they do. At the same time, mechanical LOGO skills are being developed which can be used to attack more sophisticated syntactic and linguistic concepts. Some of these possibilities will be discussed in next month's column.

Glen Bull is a professor at the University of Virginia, and teaches LOGO courses at both the graduate and undergraduate level.

New ABC's Announced

Many of you have received your copy of the NLX ABC procedures for the Apple LOGO versions. Here is some important news for you. There is now a "new improved" set of ABC's available! Many subscribers have helped point out ways to make the ABC procedures even better. If you received your ABC listing before March 12, then you probably have the "old" version. To get a copy of the new, send a stamped self-addressed envelope to NEW NLX ABC's, Box 5341, Charlottesville, VA 22905.

Creating a LOGO Environment

by Tim Riordon

(Editor's note: This is the final section of a 2-part reprinting of Tim Riordon's significant article which originally appeared in the November 1982 issue of The Computing Teacher. Reprinted with permission.)

Throughout the day, the teacher is seen consulting a list with eight student names. When a few minutes are available, the teacher looks at the list and selects one student for an impromptu 5-minute conference. Tomorrow, another eight names will be added to the list.

The first question in these conferences always seems to be, "What are you doing with LOGO this week?" Some students have elaborate plans for games or procedures. Other students, exhibiting a very different manner of working, explain that they are "messing around" with one procedure. By "messing around," they apparently mean to work with one procedure, make a change in it, run it, make antoher change, run it again, etc.

Evidently, the teacher does not try to coerce students into adopting another method. "Messing around" with LOGO procedures is valued. The teacher is sensitive to and accepting of different intellectual styles.

The second set of questions is usually, "Have you had any unexpected results? Do you have a theory about what caused this? What did you try in attempting to solve the problem?" The teacher is using questions to help students focus upon their own theorizing, and is promoting learning by having students theorize, test and revise, instead of always having them learn by direct instruction.

The central question in these short conferences always seems to be, "Is there anything you are having trouble with on your current LOGO project?" If the answer is yes, the student is asked to draw what happens and if s/he has any ideas about what might be causing the problem.

Frequently, the problem is complex enough that the student and teacher agree that they need to watch what happens on the computer. Somehow, a few minutes at recess, lunch, or after school are found so that the problem can be demonstrated.

Sometimes the teacher provides the student with an answer. Other times the teacher will give hints and encourage the student to try another approach to solving the problem. Often the teacher will tell the student to use the "masking tape strategy." Using this strategy, a problematic design is examined and then taped on the floor. The student plays turtle and walks through the procedure.

Using LOGO language, the student describes everything s/he is doing to walk the design. Sometimes the teacher or another student copies down each statement. The result is usually the LOGO procedure needed to create the design in question.

Sometimes students will say that they are done with a project or are bored with it. With some students, the teacher's resources for project ideas are offered as possibilities: "LOGO for the Apple II" by Abelson, and "Final Report Brookline LOGO Project, Parts II and III" by the MIT LOGO Group. With other students, the teacher encourages and occasionally directs that something specific be done - usually a project that the teacher believes will open up new LOGO vistas for that particular student.

"Bill, I'd like you to do a little project that makes the computer draw umbrellas of increasing size all over the screen. Do you have an idea about how you would do that?" (Bill understands how to use variables and recursion, but has not used the two ideas together; the teacher makes sure that Bill has some idea about how to begin so he will not experience great frustration.)

"Katie, would you rewrite this procedure that always draws 9-pointed stars the same size? Make it so that you can input both how big and what color the 9-pointed star is. Do you have some ideas about where you would change this procedure?"

"Matthew, I think that you will be able to solve your problem if you try the project on the wall that asks you to use a circle to make a slinky design. If you do that project and still can't figure out how to do what you want in your current project, come on back and see me."

After watching and listening to these 5-minute interactions throughout the course of a week, one can draw several conclusions. The teacher:

-is sensitive to whether or not students are headed for frustration.
-has in mind a sequence of LOGO programming concepts and a large repertoire of LOGO project ideas.
-often considers his/her teaching role not as a repository of answers but as a midwife helping students discover answers by theorizing about problems and unexpected results.

Using all of this knowledge, the teacher exercises the art of teaching in quickly discerning how to respond to each student as an individual. Some need an answer and some just a hint; some need a practice project; and many need nothing but more computer time.

continued on page 8

LOGO Environment continued

A LOGO environment has many of the attributes of a democratic classroom. Authority is distributed; sharing and cooperation are promoted; students look to their classmates as legitimate sources of information; because students can make project choices, variety rather than uniformity is the norm; rewards are intrinsic; differences in working style are valued. There is a sense of shared learning - students may hear their teacher respond, "I'm not sure how to make the computer do that. I'll work on it at home tonight."

WHAT FEATURES ARE CONSPICUOUSLY ABSENT FROM A LOGO ENVIRONEMENT?

There is no evaluation of student learning prepared by the teacher and given to students or parents. There are very few required projects which students are expected to do. Many of those that are required are selected specifically for an individual student rather than for the whole class.

While there is no formal evaluation of a child's LOGO learning, parents are kept abreast of what their kids do, sometimes through a parent night, where students demonstrate their programs. LOGO lends itself to class projects, booklets and calendars with LOGO graphic designs by each class member. It should also be noted that LOGO excites students enough to go home and volunteer information to parents on " what we did in school today."

WHY SHOULD SUCH AN ENVIRONMENT BE SET UP IN A CLASSROOM?

Why should classroom instructional time be used by students in an activity which is not formally evaluated? The most complete answer to these questions is to be found in Seymour Papert's book, MINDSTORMS: Children, Computers and Powerful Ideas. LOGO was created because of a fundamental belief that children can learn many more important things without formal instruction.

Just as children learn verbal language without formal instruction - a vocabulary of thousands of words and a set of complex syntactical rules for putting words together meaningfully - so, too, the language of mathematics can be learned. No matter what one's theory about how language can be learned, it is a fact that children learn language when they (interact with) a language-rich environment.

A LOGO environment is a mathematics-rich environment, a context including not only a computer and walls and floors with project ideas but also how students and the teacher in the classroom interact with each other. In an environment where mathematical objects and ideas are joyfully shared, played with, discussed and encouraged, mathematical intuitions and language will be learned without excessive formal instruction.

What lies behind all of this is a Piagetian view of children who learn because they naturally make and revise theories about things they are interested in.

HOW DOES ONE BEGIN CREATING A LOGO ENVIRONMENT?

Once you begin learning LOGO yourself, introduce it in your classroom. This article was written with self-contained classroom teachers in mind. In middle schools, high schools and other situations that are not self-contained, teachers can invent other ways of organizing. Get LOGO going as a peripheral activity; that is, let your regular curriculum proceed but allow individual students to miss part of it. Students who miss instruction can be treated like students who were absent. With older students, it is possible to strike a bargain- a classroom agreement. A student working on the computer will be responsible for checking with a friend about what s/he has missed.

In the beginning, you will need to demonstrate a number of things to the whole class. The question of scope and sequence is problematic. LOGO inventors fear that if a scope and sequence is published, it will invite the belief that students should be accountable for learning LOGO programming concepts. This will inevitably lead to evaluation of student learning. This will lead to LOGO being a joyless, unnatural learning learning activity.

LOGO was not meant to be taught, and the kind of learning that occurs was not meant to be evaluated like other school learning!

It is in the spirit of LOGO that you should begin the project of creating a LOGO environment with the idea that you will try something, debug it and try again.

References
Abelson, Harold. "LOGO for the Apple II." New York: BYTE/McGraw Hill, 1982.
Adams, James L. "Conceptual Blockbusting." Stanford, CA: Stanford Alumni Association, 1974.
BYTE Magazine. Entire August, 1982, issue is devoted to LOGO.
MIT Logo Group. "Parts II and III Final Report Brookline LOGO Project." Cambridge, MA: Artificial Intelligence Publications, Massachusetts Institute of Technology, 1979.
Papert, Seymour. "Mindstorms: Children, Computers and Powerful Ideas." New York: Basic Books, 1980.

Tim Riordon is a teacher and consultant in Eugene, Oregon, and is a regular contributpr to The Computing Teacher magazine.

ISSN 0734-1717

The National LOGO Exchange

Volume 1 Number 9 FORWARD 100! May 1983

Children and the Turtle Down on the Farm

by Linda Razzano

Our turtle has been living on a farm. It has visited the cows, pigs, sheep, and horses daily, as it has maneuvered through corrals and along the farm roads. It has become the busiest and most gregarious member of our farm community.

My class of kindergarten and first grade children began to use the floor turtle in October, after having had some initial experiences using the screen turtle. They used an INSTANT program for the floor turtle that was basically the same as the INSTANT program they had used with the screen turtle, with the addition of WINK and HORN commands.

The floor turtle we've been using is the Tasman Turtle, which is larger than the original Terrapin Turtle. This larger size necessitated setting aside a good deal of floor space adjacent to the computer station which could be devoted to the turtle's environment.

The children's first use of the floor turtle was simply directing it to move to a specific object and back again. They then used building blocks to construct curved and angled channels through which they drove the turtle.

Since we had been involved in a farm life unit, a natural extension of both their turtle-driving projects and their farm studies seemed to be designing a farm environment for the turtle. Once again, the children used the building blocks and constructed corrals, fences, and barns; they also equipped the farm with our rubber animals.

Next, we introduced the turtle to the farm. It was great fun to get the turtle to wink at the pigs, and then to toot the horn while passing by the cows! The floor turtle was always operated by a pair of children: one at the terminal, and one keeping the turtle's "tail" from tangling. More excitement was exhibited for this activity than for anything else we'd done this year.

continued on page 8

Cherokee Indian Symbols Come Alive With LOGO Sprites

by Pat Reilly

Recently, my second grade class at Punahou School, in Honolulu, HI, studied the importance of symbols in the Cherokee belief system. The Cherokees attached much significance to the four cardinal directions: north, south, east, and west; the colors of red, black, white and blue (or purple); animal life, particularly bird life; and the sacred numbers four and seven.

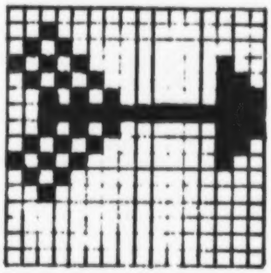
My children used TI LOGO to design some of these symbols as sprites and integrated their knowledge of the Cherokee Indians with their computer skills.

Here is the procedure the children followed:

1. Design an Indian symbol of your choice on a paper makeshape grid (16 x 16).
2. Get a makeshape number from the teacher.
3. Transfer the design to a shape on the computer using the MAKESHAPE command and the shape number from the teacher. (Arrow keys are used to designate the darkened squares.)
4. Use the design as a sprite, with a TELL and CARRY command. Set an interesting color, heading, speed, or location. It is now an Indian sprite!
5. SAVE your Indian sprite on a classroom diskette.

Here are some examples of the symbols, and some of the messages the students assembled.

red arrow: protection and life
by Jessica Diamond



continued on page 2

The National LOGO Exchange, copyright 1983 Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly September through May, $25 per year, mailed FIRST CLASS from Charlottesville, VA. Add $5 per year for addresses outside the USA. Single copy price $3. The opinions expressed by the authors are not necessarily those of The National LOGO Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress St., Salem, MA 01970. Address all other correspondence to The National LOGO Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717/83 $2.00

Editor......................Tom Lough

BACK 9

So much has happened in the last nine months! It is fun to pause for a moment to review the highlights. The shortcomings are there, too, naturally. But, as we compiled the index (see page 12) for the first year of the NLX, we experienced a feeling that something significant may have been accomplished, some needs served, some minds stirred.

Often, if we look back to where we have been, it gives us a better idea of where we are going. Of course, it is not constructive to dwell upon the past; it is worthwhile to learn from it. In particular, we want to learn ways to improve the service of the NLX.

However, when one looks back, it is sometimes difficult to see everything which is important. Distance and time can cause distortion of the things one is trying to discern.

That is why we ask for your help at this time. We feel that your points of view are important sources of useful information.

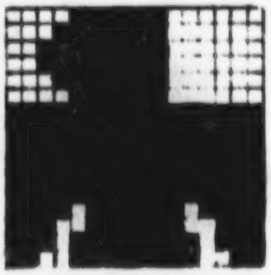
Would you please take a moment to think back about the NLX issues you have received? Then write us with any ideas or suggestions you could offer for improvement. This will help us to serve you even better.

We would like to send to each of you a very special THANKS. You have chosen to join the NLX in the first year and to grow with us. You are a special person for having taken that risk. We wish you continued success, excitement, and curiosity in your next LOGO year. We look forward to renewing our relationship with you in September. Until then,

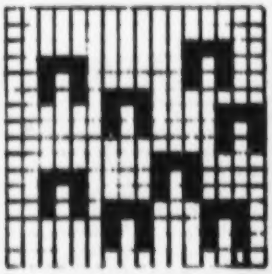
FORWARD 100!

Tom

Cherokee Symbols continued



white eagle:
peace and happiness
Klari Asato



deer tracks:
plenty of game
by Angel Vardas

The children enjoyed recalling one another's Indian sprites, combining them in subprocedures, and writing coded messages to each other. They also discovered the value of the PO (print out) TI LOGO primitive, which enabled them to change temporarily the attributes of their friends' sprites.

TELL 1 CARRY :TURKEY SETCOLOR :WHITE SETHEADING :SOUTH SETSPEED 7 gives a white eagle moving south, meaning peace and happiness.

TELL 2 CARRY :TURKEY SETCOLOR :BLACK SETHEADING :WEST SETSPEED 4 creates a black turkey moving west, meaning death.

TELL 3 CARRY :WOODPECKER SETCOLOR :RED SETHEADING :EAST SETSPEED 7 will give a red woodpecker moving east, to denote success in battle.

TELL 4 CARRY :JAY SETCOLOR :BLUE SETHEADING :NORTH SETSPEED 4 sets up a blue jay flying north, meaning trouble and defeat.

Creating these Indian symbols as LOGO sprites allowed the children to integrate their knowledge of the Cherokee belief system with their knowledge of the LOGO language. Thus, they were able to make yesterday's symbols come alive using today's technology.

(Ed. note: Dover Publications, 180 Varick St., NYC, NY 10014, has a number of inexpensive books showing many Indian designs and symbols. One such excellent ideabook is Authentic Indian Designs by Maria Naylor, Dover # 23170-4, $6.95 + .75 postage, 219 pages. A listing of 40 other related books is included. There are plenty of LOGO ideas here!!)

Pat Reilly teaches a self-contained second grade at Punahou School in Honolulu, Hawaii. She has taught also at schools in Tennessee and Georgia.

TIPPS

for

TEACHERS

by Steve Tipps

LOGO for All

Turtle graphics, printing, list processing, and arithmetic functions offer possibilities for many tasks to give computers. Few examples of LOGO are available, however. Anyone can go to a bookstore and find dozens of books containing hundreds of BASIC programs; LOGO programs are less available.

Glen Bull and I have been encouraging students and teachers to undertake computing projects using LOGO to expand the repertoire of LOGO applications. We have found that, beyond using the beginning commands, adopting and pursuing a project is the best way to learn LOGO. The demands of the project force growth with the language. Over the past two years, students in our classes have created many tasks for themsleves which demanded new skills and understandings.

One of the difficult things about creating "educational" programs with LOGO is the mind set which we bring to the enterprise. Creating drill and practice programs with LOGO is relatively simple; creating highly interactive and creative environments is demanding.

We started asking people to design "microworlds" but that label seemed to intimidate them; we dropped the label, but not the idea of using LOGO to create exciting ways of exploring ideas other than through the traditional tutorial questions and answers. Other students used LOGO for the development of teacher utilities. These examples might stimulate you to define your own projects.

BALANCE

Mary King, a principal and member of the coordinating committee for the LOGO project in Albemarle County, VA, made a balancing exploration. The turtle draws a seesaw balance with different weights on the ends. The seesaw tilts to show which side is heavier. The child has control over the position of the fulcrum and the weight placed on one end to balance the given weight on the other end. Each attempt to balance causes the seesaw to be redrawn to show the results.

Trial and error is used at first to determine the relationships needed for balance, but, as children progress, they will need and find use for the formula which describes the length times weight situation. Mary's program is not intended to replace acutal scales and weighing; instead, it extends those real situations and probably focuses directly on the relevant variables.

SKETCHING

A number of students have worked on making LOGO sketch boards. These efforts have ranged from single-key LOGO adaptations for very young children and handicapped youngsters, paddle controlled sketching pads and race tracks, and menus of shapes which can be drawn.

Sue Montgomery used her sketching program as a basis for children to create stories. As the sketch unfolded, she would stop and let the children compose a sentence using REQUEST. At the end, the child could print out the creative product and add their own version of the picture. Bernie Chesshir made a program which explores drawing in perspective.

I have worked on a single key LOGO version for 3 and 4 year olds at the UVA Child Development Center. To begin, I programmed combined distance and direction moves which are automatic responses to touching keys on the keyboard. Glen Bull chided me for combining the two types of moves into one action. We will find out how the childrn play with the computer to see if that was a bad choice. We also debated about whether we should have the children touch <RETURN> or make the keys automatic. I chose the latter. Those of you who have been through this please share your perceptions and experiences.

Of course, the questions of pre-programming LOGO is a main concern. Do you destroy the value of LOGO by reducing the child's fundamental control of processes? These projects invariably produced good learning for the programmer. When the products are used within the LOGO philosophy as examples which children can use, expand or adapt, I have fewer qualms. If they replace the exploration and understanding of LOGO, I think we have gone astray.

PIGGY BANK

Sally Gilmer created a piggy bank into which children could drop coins. The coins (TI LOGO sprites) fell into the piggy bank, beeped, and a total of all the coins flashed on the screen. She was working with learning disabled students and wanted to have a world in which they could play with addition ideas without being overburdened with calculations. The sum was responsive to the decision of the child about the kind of coin added.

continued on page 4

TIPPS for TEACHERS continued

PLANETS

Mary Ann Kivistik was working on a science unit on space. Her goal was to demonstrate how many times each planet revolved about the sun in one earth year. Text book pictures cannot show this idea; the pictures are static while the idea is dynamic. As she worked, she found more things that she would like to show with the computer. Her project will take a long time to complete, but she and the children are enjoying the development process.

Programming does not have to be done in the closet. When you are working on a project, children will give you valuable reactions and suggestions. For this project, sprites would be a perfect way of showing simultaneous movement of the planets.

FRACTIONS WORLD

Fractions are one of the most difficult ideas in elementary mathematics. Without going into a treatise on proportional thinking, I'd really like to see some work on using the computer to involve children with ratios. Programming in LOGO and scaling a drawing to one number is one way to go about this.

Jan Tuck had a wonderful car drawn using different circle parts. When she began to change parts, she discovered that all the proportions went wackeroo! In working out the correct relationships, she discovered much about ratio and geometry in a real - experimental way rather than by a rote - book way. At the end of the project, she wanted to know more about geometry and fractions.

Polly Wake has taken a different approach to fractions by displaying fractional parts of squares of differing sizes. The turtle draws squares split in half, fourths, and thirds , and colors in certain parts of the square. Many commercial programs are available which ask questions about fractional parts; in Polly's approach, children are telling the computer what kind of fractions to display, add, and subtract.

A good project on fractions needs to be done with attribute or concept formation ideas. A fractional part is displayed with the word YES or NO given. After three or four examples, the student can accept or reject a drawing as belonging. Although this edges toward drill and practice format, the content is not low level, but aimed at attribute abstraction and conceptual thinking.

OTHERS

Many other worthwhile projects are also underway. Transitional subject areas can provide exciting ways of using LOGO to explore ideas. (See page 74 of Tom Lough's article in April's Electronic Learning, for example.) Such uses of the computer should not overwhelm the legitimate LOGO problem solving, but can provide a way of using LOGO as a tool. Students and teachers can turn the projects into cooperative classroom efforts.

Students might not be able to understand every part of the program structure immediately, but the modular nature of LOGO lends itself to dividing the task up and combining everyone's parts into the final program.

UTILITIES

Teacher utilities are programs which teachers can use to do their jobs better. They are not strictly instructional, but can have instructional application. One example which Glen has demonstrated for several groups is the grade calculator. Rather than using the hand calculator for compiling grades, a simple program in LOGO asks the teacher to input the grades and it provides percentages and final average. Another teacher utility designed by Pete Davidson randomly picks students for classroom jobs.

Steve Taylor programmed LOGO to store the word lists of individual students. Each child keeps a list of words; these can represent individual spelling lists or creative writing words or vocabulary words. The word bank idea has many applications. Barb Thomas provides enrichment spelling for some of her students. She can enter extra spelling words for children to work on beyond the regular list. Both ideas point to the potential of the computer for individualizing instruction other than percentages of right and wrong.

Two students from Computer Science have been working on Senior Theses with LOGO this year. Linda Kaiser and a colleague chose to work on data base management systems in LOGO. They had much programming experience and have consulted with Keith Hammon and Terry Schwartz in Albemarle County to write student record systems. Data structures in LOGO have not received much attention in the literature. We hope that the prototype systems which are being designed will prepare us for new implementations of LOGO with more memory and features. (Ed. note: see March NLX, LOGO in the News)

continued on page 8

MICROWORLDS

by Glen Bull

Teaching Grammar to a Computer II

In last month's column, one method for teaching the computer to form plurals was outlined. The method consisted of the following approach:

```
To Form a Plural-
    If the last letter in a word
    is S, X, or Z, add ES to the
    end of the word;
    Otherwise, add S to the end
    of the word.
    End
```

This English description can be translated into LOGO in the following way:

```
TO PLURAL :ITEM
TEST MEMBER? LAST :ITEM [S X Z]
IFTRUE OUTPUT WORD :ITEM "ES
OUTPUT WORD :ITEM "S
END
```

In the past month, one fourth grade teacher has experimented with this approach in spelling class. Older children who understand list-processing instructions could write the LOGO procedures themselves. However, although these fourth graders are familiar with turtle graphics and some LOGO print instructions, they are not actually capable of writing the pluralization procedures.

AUTOMATIC CODE GENERATION

This problem was easily circumvented by use of a Text-to-LOGO Conversion unit, a powerful device which translates a procedure from English into LOGO. This capability enabled the class to describe a spelling procedure in English and evaluate the effectiveness of the procedure on a computer.

The class was already familiar with the mechanism of a procedure because of their own work with LOGO. Use of a Text-to-LOGO Conversion unit quickly became an accustomed component of the program development process, just as use of a compiler and linking loader are part of the development cycle in other systems. This Text-to-LOGO Conversion unit (you guessed it: the teacher!!) used Heuristic Evaluation Rules and proved to be more accurate and flexible than military program generation systems costing millions of dollars more!

In other words, the process of describing a procedure is just as important as the actual process of coding it in LOGO. That, at least, was the theory. In practice the class not only participated in the description of common pluralization rules, they also enthusiastically searched out arcane exceptions until it became necessary to declare a moritorium!

MORE USES FOR LAST AND BUTLAST

Previous versions of a procedure for plural formation tested for words ending in S, X, and Z. In order to test for words ending in SH and CH, it is necessary to create a module for extracting the last two letters of a word. This type of procedure may be given to a class just as the procedure for MEMBER? may be supplied for versions of LOGO in which it is not a primitive.

(Ed. note: The procedure MEMBER? was supplied in the April NLX and appears below for new readers.)

```
TO MEMBER? :ITEM :LIST
IF :LIST = [] OUTPUT "FALSE
IF :ITEM = FIRST :LIST OUTPUT "TRUE
OUTPUT MEMBER? :ITEM BUTFIRST :LIST
END
```

The strategy consists of:

1. extracting the last letter of the word.
2. extracting the next-to-last letter of the word.
3. using WORD to combine the two letters.

The following procedure is one way to implement this strategy:

```
TO LAST.TWO.LETTERS :ITEM
MAKE "LETTER1 LAST :ITEM
MAKE "LETTER2 LAST BUTLAST :ITEM
OUTPUT WORD :LETTER2 :LETTER1
END
```

When this procedure is used, it produces results such as:

```
PRINT LAST.TWO.LETTERS "JONATHAN
AN
```

This procedure, in turn, can be incorporated into a revised and updated version of PLURAL which accounts for all five cases described in most elementary texts on English.

```
TO PLURAL :ITEM
TEST MEMBER? LAST :ITEM [S X Z]
IFTRUE OUTPUT WORD :ITEM "ES
TEST MEMBER? LAST.TWO.LETTERS :ITEM [SH CH]
IFTRUE OUTPUT WORD :ITEM "ES
OUTPUT WORD :ITEM "S
END
```

This will give the following results:

```
PRINT PLURAL "BOY
BOYS
PRINT PLURAL "GRASS
GRASSES
PRINT PLURAL "WATCH
WATCHES
```

continued on page 6

MICROWORLDS continued

Of course, a number of other exceptions such as oxen and mice remain. A very intelligent computer might be programmed with a knowledge of Latin, Old English, the linguistic history of English evolution, and a number of other factors to identify these exceptions. In the distant future, such a computer might exist. There are already languages such as LISP (a grandparent of LOGO) which can give a computer a general knowledge of the world, enabling it to deduce specific characteristics.

In the interim, it is necessary to enter a list of exceptions at the beginnng of our LOGO procedure:

```
TO PLURAL :ITEM
IF :ITEM = "OX OUTPUT "OXEN
IF :ITEM = "MOUSE OUTPUT MICE
TEST MEMBER? LAST :ITEM [S X Z]
IFTRUE OUTPUT WORD :ITEM "ES
TEST MEMBER? LAST.TWO.LETTERS :ITEM [SH CH]
IFTRUE OUTPUT WORD :ITEM "ES
OUTPUT WORD :ITEM "S
END
```

RULES FOR EMPLOYING ARTICLES

The skills developed in plural formation can be used to attack syntactic rules. For example, many elementary texts on English devote a section to the use of articles. The article AN is placed before words which begin with vowels, while A is used before all other nouns. This process could be described in English in the following way:

To Article -
 -If a word begins with a vowel, use AN;
 -Otherwise, use A.
 -End

The LOGO counterpart of this rule is shown below:

```
TO ARTICLE :ITEM
TEST MEMBER? FIRST :ITEM [A E I O U]
IFTRUE OUTPUT SENTENCE "AN :ITEM
OUTPUT SENTENCE "A :ITEM
END
```

This procedure works tolerably well for many cases, but not for all instances:

```
PRINT ARTICLE "APPLE
AN APPLE
PRINT ARTICLE "PEAR
A PEAR
PRINT ARTICLE "APPLES
AN APPLES
```

The rule does not operate properly, however, in the case of plural nouns. For example, "THE BOYS" would be preferable to "A BOYS". Recognition of a plural noun is the inverse of the process of forming a plural. One useful initial assumption is that a noun ending in S is plural. Later, a separate PLURAL? procedure could be created. The revision of the English description of the use of articles could read:

To Article -
 -If a word ends in S, assume it is plural and use THE;
 -If a non-plural word begins with a vowel, use AN;
 -Otherwise, use A.
 -End

This English procedure might be written in LOGO in the following way:

```
TO ARTICLE :ITEM
TEST LAST :ITEM = "S
IFTRUE OUTPUT SENTENCE "THE :ITEM
TEST MEMBER? FIRST :ITEM [A I O U]
IFTRUE OUTPUT SENTENCE "AN :ITEM
OUTPUT SENTENCE "A :ITEM
END
```

This modification will permit the following output:

```
PRINT ARTICLE "BOYS
THE BOYS
```

The procedure for employing articles can be embedded in longer phrases:

```
PRINT ( SENTENCE [I SAW] ARTICLE "BOYS [BATTING THE BALL.] )
I SAW THE BOYS BATTING THE BALL.
```

USING LOGO TO IDENTIFY LIMITATIONS OF COMPUTERS

There are also cases in which THE is used before a noun to indicate emphasis, as in THE BALL above. The phrase THE BALL suggests that there is a particular ball in mind, whereas A BALL refers to any ball at all.

There is an enormous chasm between the first part of the rule for THE and this second part. Stated in English, we have:

1. Use THE with all plural nouns.
2. Use THE with both plural and singular nouns to indicate emphasis.

The first part is fairly easy to implement. The second part requires a considerable knowledge of linguistics, the world, and the relationship between the two. Creation of a LOGO procedure which differentiates emphasized and unemphasized nouns, perhaps through evaluation of surrounding text, would not be a realistic task.

The exercises in use of LOGO to describe grammar rules can be used to introduce computers in an English class. They also may be used to demonstrate that certain types of tasks may not be readily amenable to solution by computer. I believe that, for many, knowledge of the limits of computers may ultimately prove as important as knowledge of their capabilities.

continued on page 7

OBTAINING ADDITIONAL INPUTS

One means of determining when a noun should be emphasized is to ask the person using the computer:

```
TO ARTICLE :ITEM
TEST LAST :ITEM = "S
IFTRUE OUTPUT SENTENCE "THE :ITEM
PRINT ( SENTENCE [DO YOU WANT TO EMPHASIZE] :ITEM [?] )
TEST MEMBER? REQUEST [[Y] [YES] [SURE]]
IFTRUE OUTPUT SENTENCE "THE :ITEM
TEST MEMBER? FIRST :ITEM [A E I O U]
IFTRUE OUTPUT SENTENCE "AN :ITEM
OUTPUT SENTENCE "A :ITEM
END
```

This procedure asks whether a word should be emphasized. A sample output might resemble the following:

```
PRINT ARTICLE "APPLE
DO YOU WANT TO EMPHASIZE APPLE?
NO
AN APPLE


PRINT ARTICLE "TURTLE
DO YOU WANT TO EMPHASIZE TURTLE?
YES
THE TURTLE
```

Let your imagination go. I'm sure you can think of many improvements and extensions of these ideas. Keep us posted on what you do!

TURTLE EMPHASIS
a preliminary report

As indicated above, I want to emphasize TURTLE because we have just had our first encounter with THE Tasman Turtle, as described in the March issue of the NLX. It is a well constructed two-wheeled plexiglass hemisphere, connected to the computer through an umbilical cord of ribbon cable which carries both power and control instructions.

Our version is configured to interface with an Apple computer, and can be operated through both MIT and LCSI LOGO and BASIC. With LOGO, the usual turtle commands are simply preceded by a T:

```
TFD 100 TRT 90 etc ...
```

I set up the Tasman at home initially - I couldn't wait until I got to the office! It took about five minutes to unpack and connect the turtle. It worked the first time it was turned on, exactly as described in the manual, without difficulty.

It was easy to control the turtle from the computer keyboard. Soon, it blinked both its eyes (green lights on the front), honked its horn and rolled around the floor.

One observation I made right away was that the connecting cable was shorter than I had expected. This limits the exploration of mazes of books and other objects placed on the floor.

A touch sensor encircles the turtle, enabling it to tell when it contacts an object in front, back, or on the sides. It took just a few moments to program the turtle so that it responded to a touch on any side by retreating, blinking its eyes all the while. In the absence of any touches, the turtle was instructed to make short random movements from time to time. (It seemed almost alive!)

A floor turtle offers tremendous possibilities for both a teacher and a class. There are also uncounted possibilities for the introduciton of children to the control of real-world objects through a computer. I also have fantasies about an introduction to LOGO conducted by the turtle on the first day of class. My belief is that every school system's instructional computing program should also include a robot turtle.

I have enjoyed meeting you through this column. In the September issue, I will share some more details on the many uses for a floor turtle in the classroom.

Glen Bull is a professor at the University of Virginia, and teaches LOGO courses at both the graduate and undergraduate level.

BEST COPY AVAILABLE

LOGO Congress Announced

The Asociacion Amigos de LOGO is helping to sponsor the First International LOGO Congress. It will convene in Buenos Aires, Argentina, September 16 - 18, 1983. Seymour Papert will be there, according to Horacio Reggini, conference chairman.

Check with your school systems now to see if they would help with your expenses. This would be an opportunity for exceptional professional growth!

The National LOGO Exchange is putting together a group tour to take care of transportation and lodging for those desiring to attend the conference. Departure points on both the east and the west coast are planned.

If you would like information on the tour, please send your name, summer address and telephone number to NLX LOGO Congress Tour, PO Box 5341, Charlottesville, VA 22905. We will keep you posted as the trip develops.

Turtle on the Farm continued

In addition to the children's enthusiasm, I saw significant growth in their competencies during their work with the floor turtle. Many children made tremendous strides in their understanding of directionality. Many also became much more accurate in estimating and anticipating outcomes. And for some children, recognizing letters and finding them on the keyboard was an enormous achievement.

I keep a notebook with brief notes of each child's struggles and triumphs. Almost without exception, the children made considerable progress in using the computer by the time we retired the turtle from the farm. I also saw a transfer of these skills to their work with the screen turtle. Their projects became much more precise.

A highlight of the turtle on the farm project was the completion of a class story book. After each child had a turn controlling the turtle's activities, she/he illustrated and wrote or dictated a story about the turtle's adventures. We compiled these stories into a book that has been read over and over again.

A floor turtle's environment need not be limited to a farm. Many other curriculum units lend themselves well to the turtle's exploration. Our turtle has subsequently led us into a study of mapping, and has journeyed through the solar system. The turtle has become a facilitator for learning in subject areas as well as a facilitator for LOGO learning.

I have found great value in integrating the turtle beyond math and into the areas of social studies, science, and language arts. The computer has become not a separate and distinct curriculum, but an integral part of my classroom program. And the floor turtle has been our major inspiration. It is an excellent aid for use with young children because it sparks their imagination and provides endless opportunities for enriching all aspects of their learning.

Linda Razzano teaches a K+1 combined class at Hanscom Primary School in Bedford, MA. She also acts as a LOGO consultant for neighboring schools.

TIPPS continued

PROBLMS AND HOPES

The projects which students have undertaken have revealed some persistent problems with current versions of LOGO. The first problem is that of workspace memory. Projects of any complexty run into difficulties of using and trading files on disk. I have several projects which fill workspace two and three times. The subsequent file manipulation causes the programmer problems which should not be necessary.

Mysterious bugs which bomb some versions of LOGO when memory is nearly full and at other unexplained times are problems which need to be addressed. Speed is another (though less obvious) problem. Extensive work with lists can be slower than desired. In looking FORWARD 100 to fuller implementations, we hope that developers take into consideration people who want to do semi-serious programming with LOGO. If LOGO is to become a full purpose language for everyone, some of these deficiencies must be remedied. That is one hope.

Another hope is that you, dear readers, have a good summer and continue to grow with LOGO. If your school is wise, they will allow you to have access to LOGO this summer, and perhaps attend one of the many LOGO summer activities mentioned in the NLXTRAs. Balance skill development with application to your own projects. The projects described this month may stimulate your own project ideas.

Glen and I will be working with LOGO this summer. We will offer one, two, and three credit hour courses at the University of Virginia. These are intensive experiences with LOGO. We always learn so much from the students because they constantly challenge us with questions, problems, and new ideas. The courses begin in late July. If you are interested in more information, you can write to me at the School of Education, 405 Emmet Street, Charlottesville, VA 22903 or in care of The National LOGO Exchange, Box 5341, Charlottesville, VA 22905.

Our last hope is to complete a LOGO book for you by August. The writing of the book meets Papert's ideas of learning by making mistakes. We have a laboratory manual which students have been debugging for two years. We hope that our experience will be a valuable addition for all who are interested in LOGO.

Steve Tipps is a professor at the University of Virginia, and conducts LOGO workshops for teachers throughout the eastern United States.

Teach Your Turtle to Draw Hex Signs

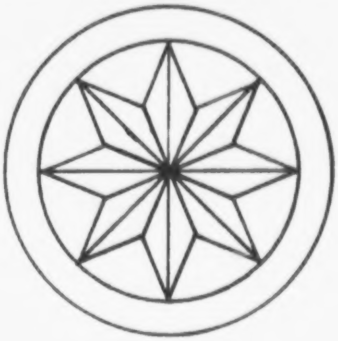
by Elizabeth Pearce

For my project in a computer course taught by Steve Tipps and Glen Bull at the University of Virginia, I developed a unit on Pennsylvania Dutch Hex Signs.

The intent of the project was to demonstrate mastery of programming skills in LOGO, including basic turtle commands, REPEAT statements, and the use of procedures and variables. But, in attempting to do that, I gained much more.

I chose to reproduce the hex signs because they are based on geometric shapes which I have always found appealing. I selected actual examples of the original Pennsylvania Dutch art form, and used a master procedure comprised of smaller subprocedures to draw each sign.

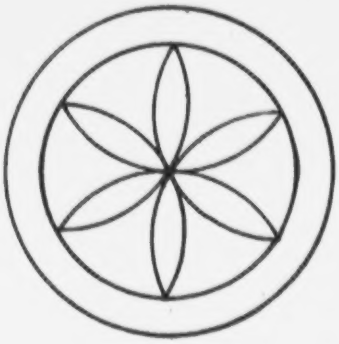
Each hex design is drawn from the center outwards. I designed my procedures so that they stop at certain places and ask for color and size selections. Variable values are computed, and that part of the design is then drawn.



I had to spend an unexpected amount of time in designing the subprocedures and defining appropriate values for the variables. But, when a design was completed, I felt a great feeling of satisfaction as the turtle moved through each subprocedure, pausing at the right places for size and color information! When I showed them to my friends, they got so excited, they broke into spontaneous applause!

It is important for us all to realize that projects such as this not only demonstrate the graphic capabilities of the LOGO language, but, more importantly, suggest a means in which the growing computer skills and interests of our students could be combined with other content areas.

These hex sign projects could be used as an independent or group project for students as a followup to a study of ethnic groups or regional areas of the United States, for example. Use of the computer as a medium for such sharing must be included in today's lesson plans.



Students might become interested in making up their own hex signs, or studying the meaning of the various components. But, such a project need not be limited to hex signs.

Early American quilt designs might be reproduced (Ed. note: see LOGO Quilting Party, NLX, December 1982). American Indian artifacts and art designs could also be studied and drawn, as shown by Pat Reilly in her article in this issue. Various architectural styles could be displayed. Map making has a universal appeal for students. Patterns in cloud formations could be illustrated.

Finding outlets for such illustrative graphics is not difficult when suggested in a manner in which students are encouraged to share their knowledge and accomplishments. Certainly, the effect of the excitement caused by a well-done project makes it all worthwhile!

I cannot wait until I have a class of my own once more, so that I can help my students develop their own LOGO projects parallel to the subjects we will be studying. And, if I have a "rainy day," I can always show them my hex signs!

References:

Folk Motifs of Pennsylvania, Frances Lichten, Hastings House, 1954.

The Pennsylvania Dutch and Their Furniture, John G. Shea, Van Nostran Reinhold Company, 1980.

Hexology: The History and the Meaning of the Hex Symbols, Jacob and Jane Zook, privately published by the authors, 1962.

Elizabeth Pearce, a graduate student at the University of Virginia, taught primary grades in Glastonbury, CT. She in majoring in Reading Education.

MECC's LOGO Book Announced

The long-awaited Apple LOGO in the Classroom by Marcia Horn and others, is ready for distribution. The book contains ten modules that enable LOGO teachers to help their students progress from LOGO fundamentals to using LOGO as a problem solving and creative thinking tool.

Teacher notes and reproducible worksheets with student activities are provided in the 217-page volume. Topics range from turtle graphics through list operations. Activities progress from familiarization to practice to self-directed, and help prepare students to explore with LOGO on their own.

The importance of student LOGO journals is discussed at the very beginning of the book. A sample is provided as a guide.

Although the book was written for use with LCSI (Apple) LOGO, most skills and activities will transfer readily to other LOGO versions. The price for Minnesotans is $16, and for others $32. Send your order to Marcia Horn, MECC, 2520 Broadway Drive, St. Paul, MN 55113.

LCSI to Release Apple LOGO Tool Kit

A disk packed full of useful utilities will be released shortly by LCSI to support its Apple LOGO. It will include procedures for printing and saving pictures, editing text, making music, low-resolution graphics use, debugging tools (STEP and TRACE) and instructions on making your own primitives.

A LOGO 6502 assembler and the commands to control the Tasman and Terrapin floor turtles are also in the package. This is what many of you advanced LOGO programmers have been waiting for.

If your students have "taken you to the LOGO limit," here is something to whet appetites. These utilities will allow extension into many areas not previously accessible.

Plans are being made to release this package through the Apple dealers. So, stay in touch with your local representative for further information. Or, write Michael Tempel at LCSI, 220 Fifth Avenue, New York, NY 10001.

TELL :SELF PENDOWN

Summer is on the way! For many of us, it promises a change from the past nine months ... an opportunity to rest, relax, read, travel, try some different kind of work.

Why not plan to use some of your summer time to reflect on your LOGO work? Further, why not write a short article for the NLX? It would be the perfect time to do so.

If you have enjoyed reading about the LOGO ideas of others, you know the importance of sharing. And, when teachers share, the children benefit. So, come on! Plan to spend a little time in the PENDOWN mode!

Articles should be about 500 words long, and include your LOGO ideas and activities, and a brief evaluation of the responses of your students. If you made use of any handy references, include information on how NLX readers could obtain them.

Send your completed article to Editor, The National LOGO Exchange, PO Box 5341, Charlottesville, VA 22905. You cannot imagine the rewards which will come your way!

New LOGO Books

Reston Publishing, 11480 Sunset Hills Road, Reston, VA 22090, has announced Apple LOGO Primer, by Gary Bitter and Nancy Watson, and 1,2,3, My Computer and Me, by Donna Beardon of the YPLA.

Turtle Enterprises North, 11515 Kathy Drive, Spokane, WA 99218, has published LOGO: A Problem Solving Approach, by Joan Davis and Joan Schenker.

McGraw-Hill's books by Dan Watt, Learning with LOGO and Learning with Apple LOGO, most likely will be out by the time you read this.

Rumors have it that Harold Nelson (Technical Editor for Microcomputing magazine and author of "LOGO: Not Just for Kids," Microcomputing, March 1982) is writing a book on advanced LOGO projects which is to be published by Addison Wesley.

LOGO Computer Systems Inc. will release their Apple LOGO Training Manual shortly. Authored by Barbara Bowen of the Apple Education Foundation and Michael Tempel of LCSI, the book will be targeted at those who present inservice training and LOGO workshops with Apple LOGO. It is organized so that a presentation can be assembled easily from the components. If you give workshops on a regular basis to diverse groups, this volume could be extremely helpful!

Save Your Apple LOGO Graphics

by Tom Lough

Since Apple LOGO (LCSI) cannot execute the DOS command BSAVE, there is no provision for saving graphic drawings on a disk. It is possible to do this, however, by "crashing" LOGO, loading BASIC into the computer without disturbing the graphics memory domain, and using the BSAVE command as in BASIC operations.

A convenient way to do this is to prepare a graphics storage disk in the following manner. Start up your Apple computer with BASIC and enter the program listing at the end of this article. Place a blank disk in the drive, and initialize it by entering INIT HELLO. (Note: When this is done, all information previously stored on the disk will be lost.)

Now, start up your LOGO system and draw something on the screen. When you are ready to save it on the file disk, perform the following steps:

1. Enter HIDETURTLE.
2. Place your initialized graphics file disk in the drive.
3. Enter .PRINTER 6 to boot the BASIC program from the file disk. You will see a confirmation that your graphics drawing is still in memory.
4. When prompted by the program, enter the name of the picture which you want to save.
5. When disk activity stops, take the file disk out of the drive, insert the Apple LOGO disk, and press <return>.
6. Resume your LOGO activity.

Later, you can print the picture from the disk using any one of many graphics dump routines available.

The name under which the drawing is saved must be selected with care. If you choose a name which is already on the file disk, the picture on the disk will be lost. If you do not know whether the name you have chosen is on the disk or not, enter * when prompted to see a catalog listing of all picture names in use on that particular disk.

Here is the program to use when initializing your graphics file disk:

```
 10  REM APPLE LOGO GRAPHICS SAVE
 20  REM COPYRIGHT 1983 TOM LOUGH
 30  REM THE NATIONAL LOGO EXCHANGE
 40  REM PERMISSION IS GIVEN TO COPY
 50  REM FOR NON-PROFIT USE PROVIDED
 60  REM CREDIT IS GIVEN TO THE NLX
 70  REM          ENJOY!
 80  REM ******************************
 90  REM INITIALIZE YOUR STORAGE DISK
100  REM WITH THIS APPLESOFT PROGRAM
110  REM ******************************
120  REM VERIFY YOUR GRAPHICS DRAWING
130  POKE -16297,0 : POKE -16304,0
140  FOR I = 1 TO 1500 : NEXT I
150  TEXT
160  HOME : VTAB (10)
170  D$ = CHR$ (10)
180  PRINT "TO SEE NAMES ALREADY IN"
190  PRINT "USE ON DISK, ENTER *"
200  PRINT "OTHERWISE, ENTER YOUR
210  PRINT "DRAWING NAME."
220  GOTO 240
230  PRINT "ENTER YOUR DRAWING NAME."
240  PRINT : INPUT N$
250  IF ASC (N$) = 42 GOTO 500
260  PRINT D$; ""BSAVE "; N$;
       ", A$2000, L$2000"
270  PRINT "DRAWING SAVED AS:"
280  PRINT N$
290  PRINT
300  PRINT "TO RESUME LOGO, INSERT"
310  PRINT "YOUR LOGO DISK, THEN
320  PRINT "PRESS RETURN"
330  GET A$
340  PR#6
350 END
500 PRINT D$; "CATALOG"
510 PRINT "PRESS ANY KEY"
520 GET A$
530 GOTO 230
540 END
```

LOGO Notes

Griff and Robbie Wigley have compiled a LOGO Resource Manual which contains 63 citations for articles on LOGO in 16 magazines, and information about 11 LOGO books, 9 LOGO-related organizations, and 13 different versions of LOGO! It is updated on a continuing basis. To get your copy, send $5 (plus .70 to cover postage) to Griff Wigley, Family Computing Inc., 918 College Street, Northfield, MN 55057. Minnesota residents add .30 for sales tax.

INSTRUCTOR magazine announces an invitation for computer using teachers to contribute tips and ideas to the expanded Bright Ideas section in each issue. This is another excellent opportunity to share your LOGO ideas with others. Compensation and a by-line are among the benefits included. Write to John Lent, INSTRUCTOR, 757 Third Avenue, New York, NY 10017 and ask for a copy of the editorial guidelines for a Bright Idea submission. Good luck!

The long-awaited newsletter of the Friends of LISP/LOGO & Kids (FOLLK) is off the presses! The first issue includes an article about FOLLK, a listing of LISP/LOGO utilities, a discussion of the role of the quote mark in LOGO operations, some interesting puzzles, and an announcement about a LOGO utilities disk which FOLLK will have available in the near future. FOLLK membership costs $25 per year for adults, and $15 for students and senior citizens. Write FOLLK, 254 Laguna Honda Blvd., San Francisco, CA 94116.

The National LOGO Exchange

1982-83 Index Volume 1

Volume 1 Number 1 September 1982

LOGO Reinforces Geometric TABS Skills by Linda Nix
Mindstorms - Don't Miss It (a review of Papert's book) by Tom Lough
Welcome! (editorial) by Tom Lough
Tipps for Teachers by Steve Tipps, Distance and Direction
SPIN (a procedure for setting a random heading)
You're Invited (announcing an opportunity for teachers to write)
Microworlds by Glen Bull, Design of a Language (chosen for reprinting in the November 1982 issue of The Computing Teacher)
Tell Congress: We Want Computers! (an invitation to support the Educational Technology Act of 1982)
LOGO Your Boat by Tom Lough (chosen for amplification and reprinting in the November 1982 issue of The Computing Teacher magazine)

Volume 1 Number 2 October 1982

A Parent Looks at LOGO by Sandy Towberman
Turtle Geometry: Don't Wait for the Movie (a review of Abelson and diSessa's book) by Tom Lough
Walls, Fences, and Camps (editorial) by Tom Lough
Tipps for Teachers by Steve Tipps, Patterns and Repetition
Microworlds by Glen Bull, The Meaning of Meaning
Fractal Fun with LOGO by Tom Lough

Volume 1 Number 3 November 1982

Hawaiian Students Love LOGO Overlays based on an idea by Elaine Blitman
Molly Watt Writes for Teachers (a review of her Creative Computing article) by Tom Lough
Hold Your Horses! (editorial) by Tom Lough
Tipps for Teachers by Steve Tipps, Patterns and Repetition II
Microworlds by Glen Bull, Is LOGO a Tool?
LOGO on Wheels by Tom Lough

Volume 1 Number 4 December 1982

Brookline Students Hunt LOGO Bugs by Jay Sugarman
Abelson's LOGO Books are Winners! (a review) by Tom Lough
Christmas and ABC's (editorial) by Tom Lough
Tipps for Teachers by Steve Tipps, Getting Started
Microworlds by Glen Bull, Languages for Implementing Solutions and Languages for Finding Solutions
LOGO Quilting Party by Tom Lough

Volume 1 Number 5 January 1983

Turtle Folders Help Third Graders by Bonnie Rifkin
YPLA Releases Turtle's Sourcebook (a review) by Tom Lough
What a Time! (editorial) by Tom Lough
Tipps for Teachers by Steve Tipps, From Turtling to Programming
Microworlds by Glen Bull, How to Use LOGO
Can the Turtle Draw a Sine Wave? by Tom Lough

Volume 1 Number 6 February 1983

LOGO and the Single Computer: How to Cope by Linda Nix
Where Does LOGO Fit In? by Bonnie Rifkin
RIGHT 15! (editorial) by Tom Lough
Tipps for Teachers by Steve Tipps, The Other Side of LOGO
Microworlds by Glen Bull, Introducing the Text Screens
NLX Pen Pals Wanted! by Tom Lough
PENDOWN, based on an idea submitted by Sandy Towberman

Volume 1 Number 7 March 1983

How to Make Your Own Sprite Graphics by Jim Fry
Burnett's LOGO Book is for Do-ers (a review) by Tom Lough
Get Ready! (editorial) by Tom Lough
Tipps for Teachers by Steve Tipps, Random Thoughts
Creating a LOGO Environment Part I by Tim Riordon
LOGO in the News by Tom Lough

Volume 1 Number 8 April 1983

Ed Emberley and the Turtle by Sharon Burrowes, assisted by David Burrowes
Tessalate Your NLX ABC's by Jim Fry
The Turtle Elite (editorial) by Tom Lough
Tipps for Teachers by Steve Tipps, Variables: More Than One Way to Tame a Turtle
Microworlds by Glen Bull, Teaching Grammar to a Computer
Creating a LOGO Environment Part II by Tim Riordon

Volume 1 Number 9 May 1983

Children and the Turtle Down on the Farm by Linda Razzano
Cherokee Indian Symbols Come Alive with LOGO Sprites by Pat Reilly
BACK 9 (editorial) by Tom Lough
Tipps for Teachers by Steve Tipps, LOGO for All
Microworlds by Glen Bull, Teaching Grammar to a Computer II
LOGO Congress Announced
Teach Your Turtle to Draw Hex Signs by Elizabeth Pearce
TELL :SELF PENDOWN by Tom Lough
Save Your Apple LOGO Graphics by Tom Lough

ISSN 0734-1717

The National LOGO Exchange

Volume 2 Number 1 FORWARD 100! September 1983

Here Comes Atari Logo

by

Shoneen Gervich

Atari Logo is on the way! Boasting four turtles, floating point arithmetic, and a full array of list processing capabilities, it has the potential to become one of the most popular versions of Logo. Developed by LCSI to take advantage of the unique capabilities of the Atari computers, it features most of the "usual" Logo primitives plus some exciting new ones which give the Logo juggling act a completely different slant.

THUNDERING TURTLES

A herd of four turtles is available to carry out both individual and collective commands. That is, each turtle can move independently or in concert with the others, responding to primitives such as TELL, ASK, EACH, and WHO.

Moreover, these turtles actually look like turtles instead of the sometimes confusing triangles. Each has a round body with four legs and a tail. A turtle head points clearly in the direction of the heading.

TURTLE CONTROLS

Each turtle has three pens, each of which can draw in any of 128 different colors. Moreover, each of these colors is available for the background, and for the colors of the turtles themselves. The Atari palette will give your students color possibilities which are practically unlimited.

The drawing speeds of the turtles can be controlled by the primitive SETSP and a number from 0 to 199. Thus, a turtle which has been told SETSP 5 will draw rather slowly, a feature perhaps useful with beginners, while one with SETSP 150 will draw much faster. Also, a line drawn at high speed will be less solid than one drawn slowly, giving an added texture capability.

The Atari Logo turtles can change shape, making it possible to create animated sequences. Up to 15 user-defined shapes may be in memory at any one time. Cars, planes, human figures, animals, and the like can come to life on the screen very easily by rapidly changing turtles from one shape to another, in a manner similar to that often done with the TI LOGO Sprites.

Possibly the most unique feature of Atari Logo is its collision detection. Beings called "WHEN Demons" check to see when specified turtles collide or when they cross lines. Predetermined lists of commands are then carried out. For example, if Turtle 1 and Turtle 3 collide, they could change shape or rebound or almost anything else imaginable.

The TOOT primitive will create sounds which can be played through the speaker of the television. Frequency, duration, and volume may be specified for two voices, making it possible to compose music, create special effects, and the like.

With Atari Logo, the heart-felt desire of almost every child to create a video game is now within reach.

NUTS AND BOLTS

Atari Logo comes in a plug-in 16K ROM cartridge which will fit the 400, 800, and 1200 models and their successors. When used with a 48K machine, such as the 800, Atari Logo has about one thousand more nodes of memory available in the workspace than any of the Apple versions.

Atari Logo comes with a reference card, and two books, "Introduction to Programming Through Turtle Graphics" and "Atari Logo Reference Manual."

A disk drive is recommended, in order to take full advantage of the language. Storage and retrieval tasks are straightforward.

The cost of an Atari Logo system is quite low. The language, a disk drive, and 64K computer can be put together for under $900. By using a tape cassette storage system and the 16K computer, you can drive the price below $400. This is about the lowest price you will find for a full Logo implementation.

EFFECTS

You may expect Atari Logo to have a significant effect on both teachers and students, and on the teaching of Logo.

The wealth of options available with Atari Logo will stimulate those already familiar with other versions of the language, and will serve as a beckoning enticement to those contemplating entering the Logo microworlds for the first time.

The multiple turtle capability makes possible new ways of thinking about solving problems. Turtle control becomes more complex, more challenging. Creativity and imagination become even more important.

Atari has made Logo affordable, powerful, and irresistible. It is up to us and our students to make the most of it.

(For more information, write: Atari Logo, PO Box 16525, Denver CO 80216 -- Ed.)

Shoneen Gervich teaches children's classes in PILOT, Logo, and (only if she has to) BASIC, through various community agencies in the South Bay area of San Francisco.

NLX Back Issues Available

A limited number of sets of Volume 1 of The National Logo Exchange newsletter are available, covering the period from September 1982 through May 1983. Many beginning Logo teachers have found this set to be a helpful resource for ideas and activities, as well as for general Logo information. To order your set, send $25 to NLX Volume 1, PO Box 5341, Charlottesville VA 22905.

From The Editor

September always has a special zing to it, doesn't it? There is something invigorating about hallways alive with young people once more. Facing a class full of new friends and new challenges is stimulating. And, of course, having Logo in your lesson plans for the coming year is exciting!

This school year is going to be one of incredible growth for Logo. Thousands of teachers everywhere are starting to use it in their classrooms. More and more versions are being released for more and more computers. Several Logo books are being published, and more are on the way. Nearly every educational computing magazine now has a Logo section.

This growth has excited many people: teachers, parents, administrators, professors, software developers, publishers. And the children are the most excited of all!

With you, we welcome this growth. Our motto FORWARD 100! was chosen to reflect this attitude. FORWARD is the direction of progress, improvement, growth. 100% of our newsletter supports you, the Logo teachers. The exclamation mark represents the excitement Logo brings to learning. By working together and sharing Logo ideas, we can multiply this growth to proportions that cannot be imagined. Your students are the direct beneficiaries of this growth!

To all our earlier subscribers, we send a hearty WELCOME BACK! We appreciated meeting you last year, and look forward to working with you again this school year. To our many new subscribers, we extend a special HELLO! We hope that this year will be one of healthy challenges and professional growth for all of us.

As a personal note, I'd like to add that, after 13 years of marriage, my wife, Posy, and I have been blessed with a child! We are pleased that our son, Kyser, will be able to grow up with Logo, and with Logo teachers such as you!

FORWARD 100!

Tom

TIPPS for TEACHERS

Borders and Frames

by

Steve Tipps

Salutations on a new year of Logo!

In this column, I enjoy presenting ideas which are not complete programs; instead, they are starters for developing problem-solving and programming skills. Teachers and students should work cooperatively as they learn from and about Logo.

In Logo, both process and product are important; in fact, they are inseparable. You should become aware of your own problem-solving and learning strategies as you switch back and forth between completing a task and playing with the capabilities of the Logo language.

One example of a project which has a nice balance between product and process is drawing frames for completed Logo drawings. Starting the new school year may mean finding interesting ways to reintroduce commands and hone programming skills in Logo. Making frames will refresh memories of commands and skills or demand new skills.

The project can be tackled by using only the beginning turtle commands of FORWARD, BACK, RIGHT, and LEFT, or it can extend and challenge skills through variable procedures with recursion. However, the focal skill for FRAMES is the use of simple procedures in REPEAT statements.

TO FRAME

The simplest frame which you can make for a turtle drawing would be to draw around the whole picture with a single turtle line. A square frame can be drawn with separate turtle commands or with a REPEAT statement, such as

```
REPEAT 4 [ FORWARD 100 RIGHT 90 ]
```

A procedure might be written for a rectangular frame. For example,

```
TO RECT.FRAME
  REPEAT 2 [ FD 100 RT 90 FD 160 RT 90]
  END
```

Different turtle canvases will require different shapes and sizes for your frames. Procedures such as CIRC.FRAME, SQ.FRAME, and RECT.FRAME can be made for fixed sizes or with variables.

Another procedure makes moving to the beginning point for the frame more convenient. SET.UP moves the turtle to the lower left corner of the frame.

```
TO SET.UP
   DRAW
   PENUP
   BACK 60 LEFT 90
   FORWARD 100
   PENDOWN
   END

TO SET.UP :DOWN :SIDE.LEFT
   PENUP
   HOME
   BACK :DOWN LEFT 90
   FORWARD :SIDE.LEFT
   PENDOWN
   END
```

The value of having a procedure for moving the turtle to the starting position is soon obvious. The variable procedure for SET.UP also demonstrates the real advantages of variability. Whether you use DRAW or HOME as the initial positioner for the turtle will depend on whether you want to redraw the canvas you are framing.

BEGINNING DECORATIONS

Decorating the frame is an obvious extension of drawing a plain frame. Changing PENCOLORs is one way to decorate, but more interesting possibilities soon arise. Gradually increasing the size of basic rectangles, squares, diamonds, or circles drawn around a picture can give the illusion of depth. These concentric shapes require repositioning the turtle from one starting corner to the next. This could be a major puzzle. The SET.UP procedure with variables may provide a clue for solving this.

Another way to decorate the frame is to place a small figure at each corner. A star, rosette, or a square in a procedure could be added into a REPEAT statement for the frame. The size for each of the decorations has been confined to 10 turtle steps or so to keep them in proportion with the screen and pictures. You and your students may want to make them larger or smaller.

```
     TO ROSETTE
        REPEAT 10 [ FD 10 BK 10 RT 36 ]
        END

     TO STAR
        REPEAT 5 [ FD 15 RT 144 ]
        END
```

By adding the rosette in the REPEAT statement for a square frame, each corner is decorated. However, the same process with STAR is not as successful at first.

```
     REPEAT 4 [ RT 90 FD 100 ROSETTE ]

     REPEAT 4 [ RT 90 FD 80 STAR ]
```

The problem encountered with the STAR is that it does not sit at the corner of the frame. Some repositioning is needed to make a CORNER.STAR for the frame procedure.

```
     TO CORNER.STAR ; REPOSITIONS STAR INTO
CORNER
        LEFT 72 STAR RT 72
        END

     TO STAR.CORNER.SQ.FRAME
        SET.UP
        REPEAT 4 [ RT 90 FD 80 CORNER.STAR ]
        END
```

By adding a decoration to the REPEAT for a rectangle or circle, a procedure like STAR.CORNER.RECT.FRAME or STAR.CIRC.FRAME can be made. STAR.CIRC.FRAME reminds me of "Mirror, mirror, on the wall," and might inspire a completely different Logo project complete with text.

CONTINUING DECORATION

When everyone sees how decorative elements can be incorporated into a REPEAT, the scene is set for many borders using different designs. Spirals, squares, triangles, points, slanting lines, lattice work, waves, and Greek keys can be added to the simple frame. For convenience, all the examples here are scaled to fit work with squares of size 10. However, variable procedures also can be developed so that sizes can easily be changed. Some stock designs which are useful might be:

```
     TO SQ ; DRAWS THREE SIDES OF SQUARE
        REPEAT 3 [ FD 10 RT 90 ]
        RT 90
        END

     TO TRI ; DRAWS TWO SIDES OF EQUAL
TRIANGLE
        RT 30
        REPEAT 2 [ FD 10 RT 120 ]
        RT 90
        END

     TO KEY ; DRAWS A GREEK KEY DESIGN
        REPEAT 2 [ FD 10 RT 90 ]
        FD 5 RT 90
        REPEAT 3 [ FD 5 LT 90 ]
        END
```

These designs can be used in REPEATs to make an edge

```
     REPEAT 10 [ KEY ]

     REPEAT 8 [ TRI ]
```

or to make the entire frame with a double REPEAT.

```
     REPEAT 4 [ REPEAT 10 [ SQ ] RT 90 ]

     REPEAT 4 [ REPEAT 10 [ KEY ] FD 10
LT 90 ]
```

The frame can also be made from combinations of elements. The combinations can be used on either side to draw different borders.

```
     REPEAT 10 [ SQ TRI ]

     REPEAT 10 [ SQ TRI SQ ]
```

For SQ, TRI, and KEY, the turtle starts in the lower left corner of an imaginary 10 by 10 grid and ends in the lower right corner. The turtle begins and ends pointing the same direction.

The drawings of ROSETTE and STAR are different because they begin and end in exactly the same position and direction (this is state transparency). ROSETTE or STAR in a REPEAT statement will draw over itself. To get one of these to make a border, a procedure which moves the turtle along the edge is needed. MOVE is a procedure to do this.

```
     TO MOVE
        RT 90 FD 10 LT 90
        END
```

REPEATs can be created with any combination of figures and moves.

```
     REPEAT 10 [ ROSETTE MOVE MOVE ]
     REPEAT 10 [ SQ MOVE CORNER.STAR MOVE ]
     REPEAT 12 [ ROSETTE MOVE MOVE TRI TRI
MOVE MOVE ]
     REPEAT 4 [ SQ CORNER.STAR ]
```

After working out combinations and positioning in the immediate mode with the turtle using single or repeated commands, an edge can be put into a procedure.

```
     TO EDGE.ROSETTE.AND.SQ
        REPEAT 8 [ ROSETTE MOVE MOVE SQ MOVE
MOVE ]
        END

     TO EDGE.STARS
        REPEAT 8 [ CORNER.STAR MOVE MOVE ]
        END
```

The edges are easier to compose with a variable for the number of REPEATs. The length of the edge can be manipulated by changing :NUMBER.

```
     TO EDGE.SQ.KEY :NUMBER
        REPEAT :NUMBER [ SQ KEY ]
        END

     TO EDGE.SQ :NUMBER
        REPEAT :NUMBER [ SQ ]
        END
```

Combinations of designs are limitless. As you find new designs, such as semicircles or lattice work, new problems of size, placement, and control will arise. Planning the length of the edge controlled by the :NUMBER in REPEAT requires measuring the pattern and multiplication of the length of the pattern.

THE FINAL FRAME

After exploring many possibilities, you may decide on a particular design, color, and size. Perhaps it is a 100 by 160 rectangular frame with squares and keys on all the sides.

```
     TO FRAME
        SET.UP
        RT 90 RECT.FRAME LT 90
        EDGE.SQ.KEY 5 RT 90
        EDGE.SQ.KEY 8 RT 90
        EDGE.SQ.KEY 5 RT 90
        EDGE.SQ.KEY 8 RT 90
        END
```

The final touch may be to add colors and corner pieces. When the frame is worked out, it can be saved (with Terrapin Logo) as a picture on the diskette with the command SAVEPICT "SQ.KEY.FRAME and read back onto the screen as READPICT "SQ.KEY.FRAME. The procedure for the turtle drawing is then executed and the whole thing can be saved as a different picture. Several picture frames can be made and saved for later pictures.

Although the original intent may have been to put a frame around a turtle drawing, making decorative borders may become a product with its own merits of problem solving and programming. The importance of pattern recognition in programming is re-emphasized for making borders. ("Tipps for Teachers, Patterns and Repetition, Parts I and II," NLX November and December 1982.)

Modular thinking is emphasized in the development of design elements which can be combined in many different ways. Positioning is separated from actual drawing by making separate procedures of MOVE and SETUP. Variability in procedures can also be introduced or explored with different size edges and frames. The project of borders and frames may also serve as an introduction to tessellations or other complex repeated graphic designs.

If you and your students develop interesting frames or other projects from frame designs, please share them with me.

Steve Tipps is a professor in the University of Virginia's School of Education, and presents Logo workshops for school systems throughout the eastern United States.

Logo Notes

UNICOM now has a version of Terrapin Logo which will run on Corvus hard disk networks. Any size Corvus disk may be used. This Logo version is rumored to perform more smoothly than the floppy disk version. For more information, write to UNICOM, 297 Elmwood Avenue, Providence, RI 02907.

Plans are being made for a National Logo Conference during the summer of 1984. We will keep you posted as more information becomes available.

MICROWORLDS

The Utilitarian Turtle

by
Glenn Bull

Logo is most commonly thought of as a tool for learning. However, the language also lends itself to development of short utilities to aid teachers in their administrative work as well. A Bachelor's thesis exploring some of these possibilities was recently completed at the University of Virginia. (Linda Keiser, "A Logo Software Package for Elementary School Teachers," Bachelor's Thesis in the School of Engineering, University of Virginia, March 1983)

Recently, when the batteries in my calculator died, I was not interested in the abstract applications of Logo. I wanted to complete the scores for a test without a trip to an all-night drug store. The ideal solution would have been a program to send the turtle to the store for more batteries! Failing that, I decided to let Logo calculate my grades.

I had foolishly written a test with 37 questions, rather than one with a nice round number such as 20 or 50. Therefore, my immediate need was a procedure to calculate percentages. This can be expressed in English as:

(Number Correct / Total Points) * 100

This quantity is rounded to a whole number in the Logo procedure which calculates percentages.

```
TO % :NO.CORRECT :TOTAL.POINTS
   OUTPUT ROUND (:NO.CORRECT /
:TOTAL.POINTS * 100 )
   END
```

The first student answered 33 of 37 questions correctly. When I typed in the percent sign followed by the number of correct answers and the total points, the following appeared:

```
% 33 37
RESULT: 89
```

It immediately occurred to me that it was silly to type in the number of points on the test for each student. Also, I usually subtract the number missed from the total number of points in my head when I use a calculator. I decided to design a procedure so the computer could perform these calculations for me using Logo.

```
TO MISSED :NO.MISSED
   PRINT % ( 37 - :NO.MISSED) 37
   END
```

Thus, I could now type MISSED followed by the number of questions which the student missed and get as a result the percent of the number correct.

```
MISSED 4
89
```

I soon tired of typing MISSED over and over. Therefore, I developed another procedure in about the same amount of time that it would have taken me to finish calculating my scores and go to bed. (Working with Logo is often like that, isn't it?)

I decided to make the number missed a value input from the keyboard. In the MIT Logo versions (Terrapin, Krell), this is accomplished with the command REQUEST, as shown in the procedures which follow. In the Apple (LCSI) version of Logo, the command READLIST is used. A set of procedures written in Apple Logo appears at the end of this column.

```
TO CALCULATE.SCORES
   MAKE "NO.MISSED FIRST REQUEST
   PRINT % ( 37 - :NO.MISSED ) 37
   CALCULATE.SCORES
   END
```

I then figured all the remaining percentages for the class, and went to bed. A few days later, I loaded the file containing the procedures to calculate scores for a different test.

When I began using the program, some of the values seemed a bit strange. I recalled that this test had 28 questions, rather than 37, as before. I changed that value in the procedure, but then discovered the results were still incorrect. After some cogitating, I realized that I had been putting in the number of correct answers rather than the number missed. Obviously, the procedure needed some more work to protect me from myself!

The better documented version has a procedure called FORMATTED.SCORES which initializes the total number of points on the examination. This procedure then calls a second procedure SCORE.LOOP which asks for the number of points missed by each student. The information which is requested is clearly labeled on the screen.

```
    TO FORMATTED.SCORES
       CLEARTEXT
       PRINT [ HOW MANY POINTS WERE ON THE
TEST? ]
       MAKE "TOTAL.POINTS FIRST REQUEST
       CLEARTEXT
       SCORE.LOOP :TOTAL.POINTS
       END
    TO SCORE.LOOP :TOTAL POINTS
       CURSOR 5 10
       PRINT1 [ '  NUMBER MISSED:  ' ]
       MAKE "NO.MISSED FIRST REQUEST
       CURSOR 5 12
       PRINT1 [ 'PERCENT CORRECT:  ' ]
       PRINT % (:TOTAL.POINTS - :NO.MISSED)
:TOTAL.POINTS
       SCORE.LOOP :TOTAL.POINTS
       END
```

the following appears on the screen:

```
    NUMBER MISSED:
  PERCENT CORRECT:
```

As each value is typed beside NUMBER MISSED the grade is displayed beside PERCENT CORRECT. Fool-proof at last! (????)

Apple (LCSI) Version:

```
    TO FORMATTED.SCORES
       CLEARTEXT
       PRINT [ HOW MANY POINTS WERE ON THE
TEST? ]
       MAKE "TOTAL.POINTS FIRST READLIST
       CLEARTEXT
       SCORE.LOOP :TOTAL.POINTS
       END
    TO SCORE.LOOP :TOTAL.POINTS
       SETCURSOR [ 7 10 ]
       TYPE SE [ NUMBER MISSED: ] CHAR 32
       MAKE "NO.MISSED FIRST READLIST
       SETCURSOR [ 5 12 ]
       TYPE SE [ PERCENT CORRECT: ]     %
 ( :TOTAL.POINTS - :NO.MISSED) :TOTAL.POINTS
       SCORE.LOOP :TOTAL.POINTS
       END
```

Glen Bull is a professor in University of Virginia's School of Education, and teaches Logo courses at both the the graduate and undergraduate level.

Slow Turtle Moves Clearly

by

Tom Lough

When introducing LOGO to older children and adults, teachers might choose to omit some or all of the usual off-computer movement activities, and let them work at the computer after a minimum of orientation. This is not necessarily bad.

However, it is important to recognize that incorrect and confusing concepts of distance and direction can be "discovered." Try the following sequence:

```
HOME
FORWARD 50
FORWARD 500
FORWARD 5000
```

"Hey! The turtle moves a shorter distance when I entered the larger numbers! I don't understand."

```
RIGHT 1
RIGHT 100
RIGHT 300
```

"Why does the turtle turn left when I told it to turn right?"

In each of the situations above, the speedy turtle assumed its new position or heading in a twinkling, leaving the student with the impression that it moved by the shortest way.

Since the screen is about 200 units high, FORWARD 500 will "wrap around" twice plus a little bit. A sudden turn 300 degrees to the right looks just like a 60 degree left turn. Thus, the turtle can appear to go backwards with FORWARD (if its heading is exactly vertical or horizontal), and to turn left with RIGHT. This can be confusing at best.

Here is a technique which I have found successful. Try using the following procedures instead of the regular ones as the very first commands given for LOGO exploration. The turtle will slow down to a speed which will allow the user to observe each complete movement directly. This means confusion such as that shown above can be avoided.

(If you are concerned that your children "putter" with large numbers in a disorganized manner, these procedures will encourage the initial use of smaller input numbers to the moves and turns. Who wants to wait all day while the turtle does RIGHT 10000?)

Here are the procedures:

```
TO F :N
   REPEAT :N [ FORWARD 1 ]
   END

TO B :N
   REPEAT :N [ BACK 1 ]
   END

TO R :N
   REPEAT :N [ RIGHT 1 ]
   END

TO L :N
   REPEAT :N [ LEFT 1 ]
   END
```

Once you have these procedures in the workspace, your students can explore with commands such as F 100, R 72, and the like. What really happens for F 500 and R 300 can be observed clearly.

This set of procedures makes it less necessary to teach degrees directly. Instead, degrees can be experienced before the idea of degrees is even conceived. Then, when the questions about the "magic" numbers of 90 or 60 or 72 arise, the concept of 360 degrees can be understood in terms of something the students have done, instead of something they have read or heard.

When you start to hear comments such as, "I wish the turtle could draw faster," you know it is time to give the varsity commands of FORWARD, BACK, LEFT, and RIGHT, with the assurance that the students have a fairly good mental picture of the turtle's motion.

I have used this technique with both children and adults. In all cases, the concepts of the turtle movements (distance and direction) seemed to be clearly and quickly understood.

In the case of adults, it gives them a boost to be able to deduce that the numbers on the turns correspond to degrees! I have seen a 12 year old boy turn an admiring face to his mother and exclaim, "Wow, Mom! How did you figure that out?!"

F 100!

Join The Logo Penpal Network

Are you looking for ideas to spice up your Logo classes? Then consider participating in the NLX's 1983-84 Logo Class Penpal Network. It is an opportunity for Logo classes in all parts of the country (and the world) to correspond with each other and share Logo ideas.

Participating Logo teachers and their classes will be matched with counterparts from various regions of the USA or Canada (or perhaps from one of several overseas countries). Assignments will be based on grade level, Logo experience, interests, computer compatibility, and Logo version(s) used.

During the school year, Logo ideas, procedures, and projects can be exchanged periodically. Your students will learn Logo, but they will also learn about people in other places. This adds an exciting new dimension to many different subject areas!

To obtain a free application kit for you and your Logo class, send a long stamped self-addressed envelope to NLX, Attn: Logo Penpals, PO Box 5341, Charlottesville, VA 22905.

The National Logo Exchange, copyright 1983 by Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly, September through May, $25 (US) per year, mailed FIRST CLASS from Charlottesville, Va. $5 (US) additional per year for addresses outside the USA, Canada, and Mexico. The opinions expressed by the authors are not necessarily those of The National Logo Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress Street, Salem, MA 01970. Address other correspondence to: The National Logo Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor.....................................Tom Lough

ISSN 0734-1717

The National LOGO Exchange

Volume 2 Number 2 FORWARD 100! October 1983

LOGO Mess Ups: Process vs. Product

What kind of a model do we provide for our students? I used to protray the image of the all knowing teacher. I knew how to read better than my students, remembered how to work math problems 100% of the time, and corrected their spelling mistakes. They saw the product of 18 years of school - not the process.

I love to learn, and I mess up constantly when I'm doing it; yet I had never showed that part of me - the real me - to my students until last year.

I work with children who have been classified as "learning disabled." By the time they enter my junior high school resouce room, they have experienced six years of school failure. Some of them can no longer confront the possibility of failure, so they actively avoid doing schoolwork. This presents quite a teaching challenge, to say the least.

ENTER LOGO

Then I found out about Logo. Its constructive approach to dealing with initial failure seemed to encourage students to learn by using it. Just what I needed. I decided to try it.

We began learning Logo together last September. I usually stayed ahead of my students in skills because I had a computer at home to practice on. Frequently, though, they asked me questions I couldn't answer right away. I had to look up the information or ask a colleague who had worked with Logo for several years. My students heard me say, "I don't know," many times, watched me rummage in books for answers, and knew that I had to ask others for help.

Logo changed my image. Now my students know that I learn the same way as they do. They place the same demands on me that I place on them: to try to achieve an attainable goal.

I want my students to experience the joy of learning. I want them to choose meaningful goals and learn to overcome those inevitable obstacles. I believe this pursuit of learning is taught best by example.

TURTLE OPERATIONS

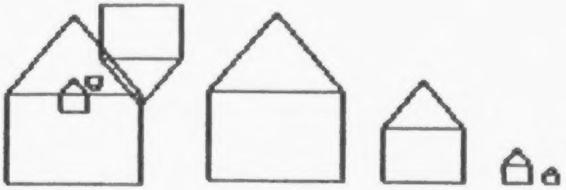
For instance, when we couldn't figure out how to program our robot turtle's touch sensors, my students saw me call M.I.T. for the information. We also operated together on our cybernetic pet's motors. How can I maintain a flawless appearance when I'm on my knees before them, gripping a screwdriver, reciting, "Righty tighty, lefty Lucy," to remember which way to turn the darn screw?

Additionally, they've seen me remain ignorant. Our robot turtle had to be told "TRT 115" for a 90 degree turn, because I had put off learning the assembly language routine that would cure it. And I knew my students would be disgusted with me when they returned to school and found out I hadn't fixed it yet. They'll pressure me to quit procrastinating, and then I really will have to do something about it!

CRAZY HOUSES

On another occasion this past winter, my students succeeded in writing a Logo procedure for the screen turtle to draw a house. I then asked them to begin creating a superprocedure with it.

That evening at home, I decided to create a superprocedure of my own to draw a row of houses of different sizes. As usual, I messed up at first. But, instead of discarding my mistake, I saved that procedure and showed it to the class the next day, along with my final product.



We all laughed at my crazy neighborhood with houses upside down and on top of each other! I can't say for sure, but I think it finally dawned on them that I really do learn the same way they do. And I realized how unfair it had been for me to show them only my final products (since my mess ups occurred at home), when their mess ups were available for public scrutiny on the classroom monitors.

SEPTEMBER STARTUP

When I decorated my room in preparation for the beginning of the 1983-84 school year, I wanted to build on my experiences from last year. I put up copies of both my original and final Logo "neighborhood." To provide my English students with a writing model, I put up my drafts of this NLX article: pages of printouts, paragraphs scribbled out with a red pen, sentences scrawled vertically in the margins.

Next to the drafts, I stapled my final copy. If that one makes it to the presses, I'll display the published version. Thus, both the final product and the sometimes painful process are on display for all to see.

By the way, I really meant to work on that robot turtle, but I waited too late. On the first day of school, James embarassed me in front of the class by asking me if I had fixed it. I said, "No, but I'll try ... if you will help me."

Thus, accompanied by his beautiful grin, we began this school year by confronting and solving a problem together, and thereby set a pattern to follow for the rest of the year.

Nancy Sopp is a special education teacher in the Fairbanks (AK) North Star Borough School District, and has published articles in national educational computing periodicals.

From The Editor

An elementary teacher told a thought provoking story at a meeting I attended recently. She had been using a word processor with a young girl in her writing class. The student had made several mistakes, and felt badly about them. As a result, she grew more frustrated.

Thinking that a demonstration might help the situation, the teacher entered a few sentences into the computer. However, she pressed some of the wrong keys quite by accident. The student suddenly grinned, pointed to the screen and shouted, "You messed up, Teach, just like me!!"

From that point, there was no holding her back. It was OK to make mistakes. Her teacher had done it, and so could she.

What can we learn from this situation? Perhaps we could get a better understanding of the value of the learning process, compared with the learning product, as seen by the student. This idea is key to the Logo learning philosophy.

As Nancy Sopp points out in her article in this issue, students usually see in the teacher the product of 18 (or more) years of education and do not consider the processes involved to attain that station. Hence, it is not obvious to them that teachers learn in the same way students do - by messing up.

"Ah, but if my students see me make mistakes, what will they think of me?" This is a sensitive question for many, and understandably so, given the traditional role of the teacher. For years, the teacher has been viewed as the great oracle, from which all wisdom and knowledge flows in a fountain to be soaked up by fledgling student sponges. If a teacher dared to utter, "I don't know," to a student's penetrating question, or tried a new kind of activity which was not effective, he or she might be branded for life!

Logo gives an opportunity to change this teaching style to a more natural one, in which the teacher becomes a guide to assist the students. But, as importantly, the teacher becomes an unashamed learner, working alongside the students.

For some, such a change is not possible. Personalities, habits, self concepts combine to prohibit innovative behavior. For others, the challenge to relax a little and not be afraid to "mess up" might be taken on successfully.

In that case, students would be able to see more of the process of education in their teachers and might become more responsible participants in their own learning processes.

Shall we give it a try?

MESSUP 100!

Tom

Calling All Turtles!

Are you a member of the Young Peoples' Logo Association? If not, you should find out about this upbeat organization. They offer many publications and activities which are of value in the classroom, including a software exchange for members. All popular computer languages (including Logo) are encouraged, but learning in the Logo style is emphasized. To obtain more information, write Jim Muller, Young Peoples' Logo Association, PO Box 855067, Richardson TX 75085.

Logo and Drive 2

Rob Muir of Claremont CA sent this tip for using both disk drives 1 and 2 with MIT Logo (Terrapin/Krell) on the Apple computer. The command .DEPOSIT 43624 followed by a 1 or a 2 will do the job. He also contributed this set of utility procedures for other NLX readers.

```
TO SETDISK :DRIVE.NUMBER
   IF ANYOF ( :DRIVE.NUMBER < 1 )
(:DRIVE.NUMBER > 2 ) THEN MESSAGE ELSE
.DEPOSIT 43624 :DRIVE.NUMBER
   END

TO MESSAGE
   PRINT [ SETDISK REQUIRES 1 OR 2 AS
INPUT ]
   END

TO DISK
   PRINT SENTENCE [ LOGO IS CURRENTLY
USING DRIVE ] .EXAMINE 43624
   END
```

TIPPS for TEACHERS

by
Steve Tipps

The Truth about Numbers

The emphasis on Turtle graphics has often obscured the arithmetic and text capabilities of Logo. Some people do not know that Logo can perform arithmetic operations. Logo comes with the basic arithmetic operators and trigonometric functions; procedures can be written to do many other operations.

Another feature hidden within the arithmetic operations is the ability of Logo to detect equality and inequality between numbers. This might be a very useful tool for teachers who wish to have children explore number concepts.

WHAT ARE THE RESULTS?

Logo performs the four major arithmetic operations of addition, subtraction, multiplication, and division. In MIT (Terrapin and Krell) Logo, Logo signifies the calculations with RESULT: and the answer. Apple Logo is a little less friendly with I DON'T KNOW WHAT TO DO WITH and the answer. Both RESULT: and I DON'T KNOW WHAT TO DO WITH are signals that a calculation has been performed and the answer has not been used to do anything. The result can be passed as input to some other Logo command such as FORWARD or PRINT or RIGHT.

The usual plus (+) and minus (-) signs are for addition and subtraction. The star or asterisk (*) is used for multiplication and a slash (/) means division. Here are some examples of arithmetic:

```
3 + 5
RESULT: 8
9 * 20
RESULT: 180
14 - 300
RESULT: -286
1000 / 33
RESULT: 30.303
```

More than one operation can also be performed on a line. However, the results are not always what is expected or desired. Consider the following:

```
10 - 5 * 3 + 2 / 4
```

If the calculations are done strictly in order from left to right, the result would be 4.25. However, Logo returns

```
RESULT: -4.5
```

After exploring a number of these arithmetic chains, children discover the arithmetic precedence of operations. Logo is being used as a discovery tool to find what rules govern arithmetic. A mnemonic I have used for the order of precedence is "My Dear Aunt Sally," to indicate multiplication and division are performed first, followed by addition and subtraction.

Children will also discover the exception to this order: parentheses. Parentheses can be used in Logo just as they are in arithmetic to cause the operation within them to be performed first. It is fun to make up chains like the following, predict the answers, and compare them with Logo's results.

```
39 + 14 * 3 / 6 - 15
39 + 14 * ( 3 / 6 ) - 15
39 + ( 14 * 3 ) / 6 - 15
```

USING RESULTS

On occasion, you may need a calculation for an input to a turtle command. Logo will do the arithmetic and give the answer to FORWARD and RIGHT. For example,

```
FORWARD 100 * 2 / 3 RIGHT 77 / 9
```

moves the turtle 66.67 steps forward and then turns it right 8.55 degrees. While this sample calculation may not seem useful immediately, figuring out the third side of a 45 degree right triangle is a more common type of problem encountered. If the two sides are 60 steps each, then, by the famous law of Pythagoras, the third side would be

```
FORWARD SQRT ( ( 60 * 60 ) + ( 60 * 60 ) )
```

The turning angles for the 45 degree triangle might come as a surprise to some. Here is one procedure to do this.

```
    TO TRIANGLE.45
       FORWARD 60
       RIGHT 90
       FORWARD 60
       RIGHT 135
       FORWARD SQRT ( ( 60 *  60 ) + ( 60 *
60 ) )
       RIGHT 135
       END
```

Converting this to an all-purpose variable sized 45-degree triangle requires substitution of a placeholder variable, such as :SIZE for 60 and the addition of the placeholder in the naming line. Students may find a way to draw triangles of any size angle and side.

Arithmetic operations can also be combined with PRINT statements as children create addition problems for themselves.

```
PRINT1 [ 200 + 90 + 1 = ]  PRINT 200+90+1
```

gives the result

```
200 + 90 + 1 =291
```

This can be "cleaned up" with parentheses,

```
( PRINT [ 200 + 90 + 1 = ]  200+90+1 )
```

to get

```
200 + 90 + 1 = 291
```

The two ways of printing are the same except for a space after the equals sign. In the first, PRINT1 is used to keep the answer on the same line as the problem. In the second, parentheses are used to tell PRINT that two (or more) things are to be printed together. (For a fuller description of children's arithmetic programs, see NLX, March 1983, Tipps for Teachers.)

WHAT IS EQUAL?

In the addition problem, the equal mark was simply printed on the page. However, when the equal mark is used outside the brackets, something different happens. The equal mark in Logo serves as a detective to find out whether a statement is true or false. In the immediate mode, an arithmetic sentence can be typed in. When RETURN is pressed, Logo responds with the results of its decision: TRUE or FALSE.

```
3 = 3
RESULT: TRUE
5 + 7 = 13
RESULT: FALSE
300 * 41 = 12000 + 300
RESULT: TRUE
300 - 100 + 6 * 4 = 236
RESULT: FALSE
```

The TRUE meaning of equals is demonstrated in Logo because the numerical equality is judged. The equal sign is active in determining equality, rather than a meaningless placeholder.

LESS OR MORE?

Knowing that expressions are equal based on their values needs to be matched with recognition that they can also be unequal. Many classroom activities have been designed for GREATER THAN or LESS THAN signs to masquerade as hungry alligators or voracious sharks wanting to eat the larger number. With Logo, the simple inequality statement is tested for truth.

```
3 < 5
RESULT: TRUE
20 > 25
RESULT: FALSE
10 + 5 < 15
RESULT: FALSE
10000 > 49
RESULT: TRUE
```

Children can play with numbers of any size and in many combinations to verify the relationships based on number. The three possibilities are clearly outlined. With two quantities, one is equal to the other, it is more, or it is less. There are no other choices.

THE PLAIN TRUTH

Playing with equality and inequality signs in Logo supports learning in arithmetic. As children grow to understand that number is an abstract property or descriptor of a set, equality is seen as a fundamental idea. The equality and inequality detectors could be used for exploring beginning number concepts, place value, and operations. At the same time, understanding of equality and testing the relationships is fundamental to work in Logo with TEST, IFTRUE, IFFALSE, and IF ... THEN.

The traditional "guess a number between 0 and 25" game is a good first TEST project for children to work out. The problem scheme would begin with playing the game in class, writing down the dialog, and tracking the decisions. Then, the challenge is to make the computer pick the number and respond with clues about too high or too low. This program can be written in only 10 to 15 lines, but it requires thinking skills and provides opportunity for many creative variations.

Working with operators, functions, and inequalities in Logo should help teachers and students see the truth about numbers. They represent an exciting world within Logo, as vital as drawing on the graphics screen or printing words.

Steve Tipps is a professor in the University of Virginia's School of Education, and presents Logo workshops for school systems throughout the eastern United States.

Logo Study Available

NLX subscriber Reinhold Wappler has just completed an excellent study comparing the capabilities of the MIT and LCSI Logo versions for the Apple computer. This detailed comparison concentrates on the needs of students in grades 2 through 4. However, much of the information applies to many other grade levels also.

Reinhold has agreed to furnish a copy of his study free to any NLX subscriber who sends him a long stamped self-addressed envelope. Send your request to:

Reinhold D. Wappler
MIT/LCSI Logo Study
252 Carter Street
New Canaan, CT 06840

Reinhold also passes along the following tip for MIT Logo users. To clean up your workspace quickly, type the following sequence:

```
ED ALL
CTRL G
ER ALL
ED
```

ED ALL brings all procedures in the workspace to the editor screen. CTRL G exits the editor, leaving all the procedures in the editor buffer. ER ALL erases the entire workspace, but not the edit buffer. ED brings back all of the procedures to the editor screen. Unwanted procedures can be eliminated with CTRL K.

CTRL C will define the procedures remaining in the editor. Thus, you will have in your workspace only those procedures you want. This technique is especially useful for tidying up cluttered disk files.

MICROWORLDS

by
Glen Bull

Numbers and Lists

In last month's column, I described a procedure for calculating average test scores. Although the procedure worked, I later received a challenge to add some safeguards to make it more foolproof.

You see, I made the mistake of putting the sentence, "Foolproof at last!" at the end of the column. Your editor immediately suggested two more ways to make the program fail. My only answer is that, although I tried to make the procedure foolproof, I obviously did not make it Lough-proof!

THE MULTIPLICATION SIGN DOESN'T LIKE LISTS

The reason Tom was able to make my procedure fail has its origin in the relationship between numbers and lists. When a number is entered through the keyboard using REQUEST (or READLIST in LCSI Logo), it is acquired as a list.

[A list, you may recall, consists of anything between brackets, such as a set of numbers, words, Logo commands, or even this sentence!]

Arithmetic cannot be performed directly on a list of numbers. If you enter the following, for example,

```
MAKE "NUMBERS REQUEST
23 34 56
PRINT :NUMBERS
```

Logo will respond with

```
23 34 56
```

But, if you try

```
PRINT 7 * :NUMBERS
```

Logo comes back with

```
* DOESN'T LIKE [ 23 34 56 ]  AS INPUT
```

The reason that the multiplication sign (*) doesn't like [23 34 56] is clear. It doesn't know whether to multiply 7 times the 23, the 34 or the 56, or possibly times all three numbers. (For those of you who teach matrix operations, this situation presents some interesting possibilities.)

In order to gain the cooperation of the multiplication sign, it is necessary to identify the number in the list on which it should operate. For example, you could tell it to multiply 7 times the first number in the list, 23.

```
PRINT FIRST :NUMBERS
23
PRINT 7 * FIRST :NUMBERS
161
```

Of course, it is possible to have a list consisting of just one number by entering something like

```
MAKE "NUMBER REQUEST
50
```

Then, PRINT :NUMBER will return 50.

In that case, it would seem that the times sign could bend its rule a little. If there is only one number in the list, it should be obvious which number to multiply seven by. However, the times sign is polite, but insistent. If the number is actually a list [between brackets], it would still like for you to point out which number to perform the multiplication upon - even if there is only one number!

```
PRINT 7 * :NUMBER
* DOESN'T LIKE [ 50 ]  AS INPUT
PRINT 7 * FIRST :NUMBER
350
```

A PROCEDURE TO INPUT NUMBERS

A procedure could be written which would always extract the first number in a list.

```
   TO INPUT.NUMBER
      MAKE "NUMBER REQUEST
      OUTPUT FIRST :NUMBER
      END
```

This procedure can be used just as FIRST REQUEST (or FIRST READLIST) would be.

```
MAKE "VALUE INPUT.NUMBER
50
PRINT 2 * :VALUE
100
```

VIEWING THE OUTERMOST SET OF BRACKETS

At this point, it may be helpful to verify one aspect of Logo. Logo normally does not print the outermost set of brackets which surround a list. This is clearly seen in a PRINT statement.

```
PRINT [ THIS IS A LIST ]
THIS IS A LIST
```

The contents of the variables "NUMBER and "VALUE appear to be identical when they are printed, because Logo drops the outermost set of brackets around the contents of "NUMBER.

```
PRINT :VALUE
50
PRINT :NUMBER
50
```

However, the brackets can be seen if a list of all the global variable names is printed out. This can be done with the command PRINTOUT NAMES (PONS in LCSI Logo). When all the NAMES in the workspace are printed out, the brackets around the contents of "NUMBER can be seen.

```
PRINTOUT NAMES
"VALUE IS 50
"NUMBERS IS [ 23 34 56 ]
"NUMBER IS [ 50 ]
```

FILTERING OUT LETTERS

The procedure INPUT.NUMBER provides a framework for guarding against other potential problems. For example, Tom suggested that my procedure would complain if a letter was typed when the procedure expected a number. This type of problem can be seen in the following sequence.

```
MAKE "VALUE INPUT.NUMBER
B
PRINT 7 * :VALUE
* DOESN'T LIKE B AS INPUT
```

The reason the multiplication sign does not like the letter B is clear. But, it is not uncommon for a letter to be typed when a number is intended. For example, the letter "L" is frequently used in place of the number "1" by office personnel who are used to a typewriter keyboard. Another common error is typing the letter "O" in place of the zero "0". Since INPUT.NUMBER will only be used when a number is expected, a test to filter out letters can be inserted.

```
TO INPUT.NUMBER
   MAKE "NUMBER REQUEST
   TEST NUMBER? FIRST :NUMBER
   IFFALSE OUTPUT INPUT.NUMBER
   OUTPUT FIRST :NUMBER
   END
```

The Logo command NUMBER? (NUMBERP in LCSI Logo) asks whether the value which follows is a number. If it is not, the procedure INPUT.NUMBER is recursively executed, until a number is encountered.

PROTECTION AGAINST EMPTY BRACKETS

Although the procedure is improved by the filter to screen out letters, it still has a potentially fatal bug. To see why, it is necessary to observe the effect of the list operator FIRST on several types of Logo lists:

```
PRINT FIRST [ 23 34 56 ]
23
PRINT FIRST [ 73 ]
73
PRINT FIRST [ ]
FIRST DOESN'T LIKE [ ]  AS INPUT
```

The first item in a list of the three numbers [23 34 56] is 23. The first item in the list containing one number [73] is 73. However, what is the first item in a list that does not contain anything? The question clearly does not make sense. For that reason, FIRST will explain that it doesn't like empty brackets as input.

In the case of INPUT.NUMBER, occasionally an individual may press the RETURN key without typing either a letter or a number. When that happens, an empty list without anything in it is created. As soon as FIRST encounters the brackets with nothing between them, it delivers the error message shown above, and the procedure explodes. Try running the last version of the procedure INPUT.NUMBER, and press the RETURN key in response to REQUEST. What happens?

The way to guard against this occurrence is to insert a test to see whether the list is empty before FIRST is allowed to see the list. If the list is empty, the procedure INPUT.NUMBER is recursively executed.

```
TO INPUT.NUMBER
   MAKE "NUMBER REQUEST
   TEST :NUMBER = [ ]
   IFTRUE OUTPUT INPUT.NUMBER
   TEST NUMBER?  FIRST :NUMBER
   IFFALSE OUTPUT INPUT.NUMBER
   OUTPUT FIRST :NUMBER
   END
```

This revision would appear to be nearly both foolproof and Lough-proof, but is it? There is at least one condition in which this procedure works functionally, but is deficient from an esthetic viewpoint. This condition involves destruction of the formatted text on the screen.

If a letter is typed in place of a number, the cursor does not return to its original position. Instead, the cursor moves to the next line, destroying text to its right on the preceding line.

A partial fix can be adopted by recording the position of the cursor before asking for input. If the input is not a number, the cursor can be returned to its original position before requesting another input. The position of the cursor can be determined by examining memory locations 36 and 37 in MIT (Terrapin and Krell) Logo. The same information can be obtained through use of the command CURSOR in LCSI Logo.

This is the final version which seems to correct all the bugs with which I am familiar. However, I am sure there is at least one more. Who will be the first to find it?

```
TO INPUT.NUMBER
   MAKE "HORIZONTAL .EXAMINE 36
   MAKE "VERTICAL .EXAMINE 37
   MAKE "NUMBER REQUEST
   TEST :NUMBER = [ ]
   IFTRUE CURSOR :HORIZONTAL :VERTICAL
   IFTRUE OUTPUT INPUT.NUMBER
   TEST NUMBER? FIRST :NUMBER
   IFFALSE CURSOR :HORIZONTAL :VERTICAL
   IFFALSE OUTPUT INPUT.NUMBER
   OUTPUT FIRST :NUMBER
   END
```

The corresponding LCSI version for the Apple is

```
TO INPUT.NUMBER
   MAKE "POSITION CURSOR
   MAKE "NUMBER READLIST
   TEST EMPTYP :NUMBER
   IFTRUE [ SETCURSOR :POSITION ]
   IFTRUE [ OUTPUT INPUT.NUMBER ]
   TEST NUMBERP FIRST :NUMBER
   IFFALSE [ SETCURSOR :POSITION ]
   IFFALSE [ OUTPUT INPUT.NUMBER ]
   OUTPUT FIRST :NUMBER
   END
```

Glen Bull is a professor in University of Virginia's School of Education, and teaches Logo courses at both the graduate and undergraduate level.

MECC '83 Conference Announced

The Minnesota Educational Computing Consortium announces its MECC '83 Educational Computing Conference, scheduled for November 18-22, 1983. A series of 2-day Logo workshops will be offered, as well as Logo activity presentations. For more information, write MECC '83, 2520 Broadway Drive, St. Paul MN 55113, or call 612-638-0683.

New Mexico Students Think Logo is VAN-tastic

The Alamogordo Space Center and New Mexico State University have joined forces to produce a Logo program that is really rolling, in every sense of the word! A van called "The Computer Experience" brings Logo to elementary students and teachers at schools throughout the state.

The van is equipped with 10 TI 99/4A computers, and is staffed by two professional educators. They can set up a temporary computer facility which handles up to 30 students at a time, providing them and their teachers with a familiarization of computers in general and of Logo in particular. Once all the classes and teachers of a school have rotated through the introductory activities, the computers are loaded back into the van and "The Computer Experience" roars off to its next destination.

Perhaps this innovative concept is one which could be used in your state or local situation. For more information on "The Computer Experience," write to: Robert F. Content, Executive Director, The Computer Experience, Alamogordo Space Center, PO Box 533, Alamogordo NM 88311.

LEARNING WITH LOGO

Arrives at Last

Learning With Logo, by Daniel Watt, is a likely candidate for a "Logo book of the year" award. Its 365 pages are full of activities which are designed to help beginners learn Terrapin or Krell Logo on the Apple computer.

To say that the book has been long awaited is an understatement. Its release finally came in late May. When Watt stood up at the NLX Birds-of-a-Feather Logo Session at NECC in Baltimore and raised the book above his head in triumph, he was greeted with a sustained round of warm applause. No words were necessary.

"Learning With Logo" takes readers through turtle graphics first, with basic commands, recursion, and variables. A study of the classic POLY is made, followed by work with numbers, words, and lists. The last half of the book is devoted to extending these ideas, using special tools and reference material.

Watt drew upon his extensive Logo experience and designed his book so that readers could learn by personal exploration. He has written many procedures to facilitate this exploration. These microworlds contain important learning concepts and are easy for beginners to use.

Watt stresses that, for maximum benefit, the book should be used with a diskette containing these procedures. A listing of the procedures, from which a diskette can be made, appears in the book. Or, the diskette could be ordered for $15.95 additional.

One of his sets of procedures is an activity called SHOOT, which helps learners gain experience in estimating turns and distances. A circle is drawn on the screen and the turtle is displayed some distance away, pointing in a random direction. The learner turns the turtle until it is pointing toward the circle, and then "shoots" the turtle a specified distance forward. If the turtle stops inside the circle, a congratulatory message is displayed. Otherwise, the learner is given the opportunity to make adjustments.

Watt points out that young children need a lot of practice estimating distances and directions. Activities such as SHOOT will help give them the turtle control skills necessary for more advanced work. He also encourages more experienced Logo users to modify his procedures into more complex ones, and gives extensive hints and suggestions for doing so. It is here that the richness and depth of the book shows through.

Teachers will find these suggestions to be gold mines of ideas for other Logo projects. In addition to being a valuable reference, portions of the book could be used as an individual text for children as young as 10 or 11. A 13-year-old reported to me that about one-third of the book was "moderately difficult" to read, and the rest was easy.

An article, "Learning With Logo," on page 40 of the April 1983 issue of Classroom Computer News magazine (now Classroom Computer Learning) contains excerpts from his book. This gives a more detailed foretaste of what to expect.

Watt has four other similar books on the way for learning with Apple (LCSI) Logo, Atari Logo, Commodore Logo, and Texas Instruments Logo.

Learning With Logo, Daniel Watt. BYTE/McGraw-Hill, $19.95.

Logo Notes

Dagobert and Lissa Soergel have written a detailed comparison of the commands in the MIT and LCSI Logo versions for the Apple computer. To receive a copy, send a long self addressed envelope with 54 cents of postage attached to:

Dagobert and Lissa Soergel
MIT/LCSI Logo Comparison
College of Library Info Sci
Univ of Maryland
College Park, MD 20742

Have you ever wanted to make a printout of your LCSI Logo procedures without having the .PRINTER 0 appear at the end? Here is one way to get around it. Enter the following all on one line.

```
.PRINTER 9 ( POALL ) .PRINTER 0
```

From Four Bugs

to a

Society of Turtles

by
Michael Burke

The cover of the August 1982 BYTE magazine (the LOGO issue) depicts an Escher-like sketch of the paths of four bugs. The paths evolve from placing the bugs at the corners of a square and having each bug always moving towards its clockwise neighbor.

The LOGO program that simulates this motion uses property lists to keep track of the positions of the bugs and transports the turtle from one bug position to another to draw small portions of the path. (Ed. note: see pages 236 and 238, BYTE magazine, August 1982)

PROBLEM SOLVING WITH BUGS

While the statement of the problem and the projected solution can be simply stated, the published program is incredibly complex in comparison. The program is useless in solving similar problems. For example, what paths evolve if we have five bugs at the vertices of a five sided figure? No part of the published program can be used in this new program.

The complexity of the published program arises because of the unnecessary use of local and global variables, the MAKE command, and the entanglement of bug representation and bug motion. There is also no need to ever think of the location of the bug as a point in a cartesian coordinate system. It is the kind of solution that a progessional programmer would come up with; it is intimidating to the non-professional.

A problem solving approach leads to a cleaner solution and a powerful mechanism for dealing with similar problems. The whole philosophy behind problem solving implies that we can learn a great deal on the way to the solution.

MANY BUGS TO SOLVE A PROBLEM

A bug has the same properties as a turtle, so let's think in terms of multiple turtles. Each bug can be represented by a turtle. Let's also assume that our turtles have names: FRED, SALLY, JANE, and SAM. Finally, we assume that the turtles are positioned at the corners of a square as follows:

```
FRED        SALLY



SAM         JANE
```

The simulation in LOGO is easily specified in this setting where :FRED, :SALLY, :SAM, and :JANE are assumed to be variables representing turtles.

```
TO MOVEBUGS
   MOVEBUG :FRED FINDLOC :SALLY
   MOVEBUG :SALLY FINDLOC :JANE
   MOVEBUG :JANE FINDLOC :SAM
   MOVEBUG :SAM FINDLOC :FRED
   END
```

FINDLOC is a procedure which returns the location of the named turtle, and MOVEBUG moves the named turtle a small distance towards that location.

BUG QUESTIONS

There are several questions that need to be addressed here: What is a turtle? How do I control multiple turtles? How can I represent more than one turtle in a LOGO that only provides one?

The second question is the important one. That is, we need to direct messages like FORWARD 10 to a specific turtle. One way is to require a name of a turtle as an input to each turtle command. However, extra inputs require extra keystrokes, and we would soon tire of typing in the name if we are sending a long sequence of messages to the same turtle.

WHERE DO BUGS COME FROM?

We have not yet mentioned how turtles are created in Mattel LOGO. Actually, they are HATCHed. HATCH "FRED will create a new turtle whose name is FRED (and will associate the new turtle with the variable :FRED) at the current turtle's position.

Thus, FRED, SALLY, JANE, and SAM could be created either in calculator mode or programmatically by defining a function which hatches four turtles at the desired positions. For example,

```
TO HATCHBUGS
   HATCH "FRED
   FD 30 RT 90 HATCH "SALLY
   FD 30 RT 90 HATCH "JANE
   FD 30 RT 90 HATCH "SAM
   END
```

So what needs to be done to simulate multiple turtles, when you only have one? You need to have a representation in mind (e.g. property lists), a way to keep track of several turtles (a list of turtle names associated with a global variable) and you need to define functions that act like HATCH and ASK. The not-too-difficult details are left as an exercise in problem solving.

FOUR BUGS FINISHED

When you have accomplished this task, you not only have a solution to the four bugs problem, you have a powerful mechanism that can be employed to solve other multiple turtle problems. A by-product of problem solving should always return more than the solution to a specific problem.

The ASK primitive of Mattel LOGO suggests more than a polite way of commanding a turtle to do something. It suggests that each turtle is an independent object that will listen politely, but is free to act in any way it deems appropriate.

Mattel LOGO has multiple turtles and the right idea on how to manage several turtles. There is always a single turtle that is listening for messages. When we want to send a message to another turtle we simply ASK it to respond. For example,

```
ASK :FRED
FORWARD 50
RIGHT 90
ASK :JANE
BACK 20
LEFT 30
```

will cause FRED to move forward and turn right, while JANE will move backward and turn left. When only one turtle exists, all messages are sent to the existing turtle.

The definitions of FINDLOC and MOVEBUG are then easily designed using the Mattel primitives ASK, POSITION, FACE, and FORWARD.

```
TO FINDLOC :BUG
   ASK :BUG POSITION ?
   END

TO MOVEBUG :BUG :LOC
   ASK :BUG FACE :LOC FD 10
   END
```

FINDLOC asks the named turtle for its location. POSITION is a primitive function that requires either a list of two numbers or a "?" as its only input. If the input is a "?", then the position of the current turtle is returned. If the input is a list of two numbers, the turtle will be moved to the corresponding point.

MOVEBUG asks the named turtle to FACE the location input (its leftmost neighbor) and move forward 10.

The possibilities here are endless. Imagine a family of turtles each of which is programmed with a certain behavior (reactions to certain kinds of messages). We now have a microworld of a society. We might even be able to change the behavior of a turtle if we could communicate to it in the right way!

Michael Burke teaches graduate and undergraduate courses in mathematics and computer science at San Jose State University, and conducts LOGO workshops throughout California.

ERIC Announces Computer Newsletter

ERIC's Clearinghouse on Elementary and Early Childhood Education has begun publication of a bimonthly newsletter called "Micronotes." It contains information on applications of microcomputers for instructional, creative, recreational and administrative uses, and usually mentions several Logo activities, among other things. The cost is $8 per year. ERIC/EECE, College of Education, University of Illinois, 805 West Pennsylvania Avenue, Urbana IL 61801, or call 217-333-1386.

New Series of Logo Booklets Announced

Kathleen Martin and Donna Bearden have announced a new series of booklets which give teachers Logo based activities for use both on- and off-computer. Each booklet in the series concentrates on a separate group of mathematical concepts. The activities show students how to use Logo to explore and understand these concepts.

The first volume in the series, "The Rule of 360," contains 50 pages designed for 10 to 12 year old students who have had some Logo experience. The second volume, "Polyspi Inspi," will be available by late October. Each booklet costs $7.95, and may be purchased with a Logo-specific diskette for $4.95 additional. Diskette versions available include Apple (LCSI), Terrapin/Krell, Atari, TI, and Commodore.

For more information, write Martin-Bearden Inc., 1908 Sandy Lane, Irving TX 75060, or call 214-253-6579.

Logo Research Directory to be Compiled

The National Logo Exchange is compiling a directory of all persons conducting any research relating to Logo. If you would like to be listed, send your name, address, telephone number, position, and a brief summary of your research interests and activities. Copies of the directory will be made available in mid-1984. Mail your information to Research Directory, Attn: NLX, PO Box 5341, Charlottesville VA 22905.

Start a Local Logo Exchange

Do you feel as if you are alone in the Logo world? Would you like to be able to talk with Logo folks from other schools once in a while? Do you want to know someone nearby whom you can call for Logo help? Then think about forming a Local Logo Exchange.

Local Logo Exchanges (LLX) are made up of Logo teachers and other workers who want to meet periodically to exchange ideas and share Logo activities in an informal setting. The NLX supports the spontaneous formation of such groups by furnishing suggested guidelines for organizing, giving publicity in the NLX newsletter, and helping to identify other Logo workers nearby as prospective members.

If you are interested in organizing such a group, then write to NLX Local Logo Exchange Program, PO Box 5341, Charlottesville, VA 22905. Tell us something about your situation, and what you hope to do. We will send you a copy of the LLX Guidelines and suggestions for getting started.

What better way to learn more Logo and get to know others in the field at the same time?

Logo Summer Courses

If you are teaching a Logo course at the college or university level during the summer of 1984, please send information to the NLX as soon as possible. Our readers would be interested in the location, dates, number of hours, prerequisites, cost, and general content of the courses. We will include your course announcement in our spring 1984 issues. Send information to: 1984 Logo Courses, Attn: NLX, PO Box 5341, Charlottesville Va 229205.

The National Logo Exchange, copyright 1983 by Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly, September through May, $25 (US) per year, mailed FIRST CLASS from Charlottesville, Va. $5 (US) additional per year for addresses outside the USA, Canada, and Mexico. The opinions expressed by the authors are not necessarily those of The National Logo Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress Street, Salem, MA 01970. Address other correspondence to: The National Logo Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor...Tom Lough

ISSN 0734-1717

The National LOGO Exchange

Volume 2 Number 3 FORWARD 100! November 1983

Logo Stepping Stones for Young Minds

by

Jan Cardwell

Logo was originally designed so that its commands could be readily understood by school aged children. It has been very successful with students who can read and conceptualize symbols and graphics. But what about young children who have not yet learned to read?

Educators of the very young know that their students are eager to learn more about their environment, what makes it work, and how to control it. These children, when presented with material in the proper manner, can learn things which are not usually found in the preschool curriculum. The use of Logo is one such example.

In teaching Logo to young children at St. Anne's - Belfield Preschool, I found a clue to success. The concept of the turtle must be introduced in terms the young mind can comprehend. These terms first must make the turtle a concrete image, then clarify the concept of giving a command, and finally provide the opportunity for practice of skills (repetition).

When I did this in a series of motivating stepping stone activities, the children learned readily and easily. Additionally, they all had a lot of fun!

Here is a sequence of Logo skill building tools Etta Legner and I have used with our preschool children during the past year. With this Logo sequence, it was possible to accommodate the needs of a variety of learning styles: individually and by group; both teacher directed and student directed.

Traveling Triangle Turtle. This stuffed triangular shaped turtle becomes the children's first encounter with Logo. I initially guided their learning with questions, leading them to discovery of the movement and turn commands in a natural way. Then, as the children called out the commands, I moved the turtle for them. Early mastery of these commands with the "concrete" turtle seemed to facilitate the later transfer to the more abstract screen turtle. Also see Instructor Magazine, October 1983, page 166. For a free copy of the turtle pattern, send a stamped self addressed envelope to Jan Cardwell Turtle, St. Anne's - Belfield Preschool, 2132 Ivy Road, Charlottesville VA 22903.

Large Turtle Graphics. The traveling triangle turtle, equipped with a marker as its tail, traced its path as I moved it around on large sheets of paper mounted on a low table or on the floor. Students gave commands which I wrote down on a poster. The students counted the steps as I moved the turtle through the drawing. This idea is similar to using the robot turtles advertised, but is much less expensive!

Turtle Sandbox. Here was an opportunity for child directed learning and practice. Children moved plastic turtles around in a large tray or sandbox partially filled with sand or sawdust according to commands from each other. As the turtles moved, they left a furrow as a trace of their movement. The children copied the commands they used from a display chart. Thus, they saw the written commands as well as the graphic results in the sand. I set up our sandbox area to be used individually during center time (free choice) after a brief introduction.

Turtle Pencils. Students used pencils with paper triangular turtle shapes attached to draw on monitor screen outlines reproduced on paper. Drawings were produced according to turtle commands from me and from other students. Note that the sequence is leading up to closer association with the computer.

Stepping Stones continued

Turtle Magnets. I pasted turtle cutouts and triangular shapes onto small magnets. Then, the children moved these turtles around on metal cookie sheets. At first, I called out commands to them. Then they enjoyed giving each other commands. Later, I had the children tilt their cookie sheets to a vertical position, so they could discover that FORWARD still meant the same thing, regardless of how the cookie sheet was oriented. The transfer to the computer screen from this activity was a natural extension of what they had been doing, with little of the usual confusion of turtle orientation.

I believe that learning experiences in a sequence such as this should be presented to young children well before their first exposure to the computer. The introduction of the triangle shape on the computer screen should be the last step in a series leading from the highly concrete turtle shape (stuffed traveling triangle turtle) to the more abstract (turtle magnet).

For very young children, the added mystique of the "magic" computer turtle movements is especially motivating. It greatly enables children to personify the turtle. This makes it easier for the young to identify with the turtle when using the Logo commands and walking like a turtle.

When it comes to teaching young students about Logo and the computer, the only limit is the teacher's imagination. Success is determined by the educator's ability to bring concepts to students in a form they can appreciate and with which they can personally relate.

Jan Cardwell teaches in the preschool of St. Anne's - Belfield, a private school in Charlottesville, VA, and has published several Logo teaching tips in national educational periodicals.

NOVA Information

For information on transcripts and copies of the October 25 NOVA program which featured Seymour Papert and Logo, write to WGBH, 125 Western Avenue, Boston MA 02134, or call (617) 492-2777.

From The Editor

Over the years, my wife has given me many wonderful gifts. As I was sitting in church a few Sundays ago, the preacher said something which reminded me of a very special gift she gave me before we were married.

The sermon was on persistence, and the anecdote was about the classic race between the tortoise and the hare. (Of course, anything about a turtle claims my immediate attention!) The minister told how the hare jumped around all over the place, while the turtle persistently kept going forward a little at a time. By this point, I was leaning forward myself, in rapt attention.

As the story ended and the sermon turned toward heavier thoughts, I must confess that my mind wandered back in time to 1966. In that year, Posy presented to me a small book entitled, The Marvelous March of Jean Francois, by John Raymond. Published by Doubleday, it was barely 60 pages long.

It chronicled the adventures of a mythical drummer boy in Napoleon's army. The young lad fell under the spell of his charismatic commander's orders, "March forward, ever forward!" His dedication to these instructions led him on a fantastic journey around the world, full of excitement and interesting experiences.

The lesson of the book, of course, is similar to that of the tortoise and the hare. Problems can be solved, progress measured, challenges overcome, if we keep moving forward, ever forward.

Embedded here are implications for the Logo learning environment which merit serious thought. Isn't it a fortuitous coincidence that the drummer boy's movement was directed by the FORWARD command?

Let us all do our best to make progress toward our chosen personal and professional goals. In so doing, we will not only be coming closer to accomplishing what we set out to do; we will also be doing what the little drummer boy did for me: serving as an example of persistence for others.

FD EVER FD 100! Tom

TIPPS for TEACHERS

by
Steve Tipps

Both Sides Now

In the strict mathematical sense, symmetry refers to the exact correspondence of shapes or figures around one or more axes. A more general definition of symmetry refers to correspondence or equivalence of events or objects in everyday life.

Recognition of symmetry is a thinking skill which is first cousin to recognition of patterns. Instead of being an exact repetition, symmetry requires recognition of movements which are moved or reversed or rotated. Logo can be used to create symmetrical designs and discover the rules needed to achieve symmetry.

USING MIRRORS TO DISCOVER SYMMETRY

The simplest form of symmetry is illustrated when a mirror reflects a pattern or design. Many commercial materials are available for mirror activities with pattern blocks and task cards. The same activities can be done with small metal or plastic mirrors from the drugstore or variety store. (Avoid cheap glass mirrors because they break easily.)

When a mirror is placed upright on a page of a book, a piece of wood, a large alphabet letter, a picture of a face, or an abstract design, it reflects exactly what is there. This is sometimes called folding symmetry because the line which the edge of the mirror makes is a fold line. The images on either side are exact duplicates from the fold line out.

Whether representational or abstract, the design and its mirror image complete a pattern. If the original design has a meaning in which directionality is essential, the mirror image has the reverse (reflected) direction. A child in front of the mirror raises a left hand; the image raises its right hand.

Print the word HELP in block letters on a piece of paper. Holding the mirror vertically, place the edge on the leftmost line of the H and observe the HELP reflection.

ꟼ⅃ƎH

The design is still HELP, but the mirror image has the letters in reverse order and "wrong-side" out.

Some designs are already symmetric and have a natural fold line. The mirror can be put on the E so that half of it is covered up, but the visible part of the E and the mirror image together still form the letter. Try to find a fold line or lines for the H, L, P, S, T, etc., as well as other designs.

Students can make symmetric pictures by folding construction paper in half. When paint blobs are placed on one side of the folded paper and the other side is folded over, the result is a symmetrical "blob" picture.

String paintings can be done in the same way with the string dipped in paint and placed on one side of the paper. After the paper is folded over, the string is pulled out, leaving a symmetrical swirled design on both sides.

LOGO LINE OF SYMMETRY

Creating designs which are symmetric to a fold line entails development and refinement of many Logo skills. A vertical line drawn down through the middle of the screen divides the turtle world into into a side for a design and a side for a mirror image.

Starting at HOME, a SQUARE.RIGHT will draw on the right side of the fold line.

```
TO SQUARE.RIGHT
     REPEAT 4 [ FD 80 RT 90 ]
     END
```

This procedure also could be used to draw the mirror image on the left side by either LT 90 SQUARE.RIGHT or FD 80 RT 180 SQUARE.RIGHT. The results would be adjacent squares on either side of the fold line. The process, however, is not symmetrical. Making symmetric designs in a symmetrical manner requires skill with direction moves.

The following exercise is both fun and instructive. Hold a mirror upright on a piece of paper, as you did when observing HELP. Watch your hand in a mirror while you draw a square. Start with the pencil at the mirror/paper edge. Draw FORWARD along the edge, RIGHT 90, and FORWARD again and so on.

Tipps continued

During each move note any differences between the drawn design and the mirror image. Does RIGHT in the design become LEFT in the image? Directionality is the main change in symmetrically drawing symmetrical images.

The symmetrical procedure corresponding to SQUARE.RIGHT is SQUARE.LEFT.

```
TO SQUARE.LEFT
     REPEAT 4 [ FD 80 LT 90 ]
     END
```

A master procedure SQUARES could call both of them and draw the design and its mirror image. Any design which can be drawn on one side of the fold line can also be put on the other side. TRIANGLES, HEXAGONS, CIRCLES, and other regular polygon master procedures are made with appropriate right and procedures.

COMBINING SYMMETRICAL PROCEDURES

After creating a treasury of symmetrical designs, combine them in different combinations. Positioning moves are made along the fold line.

All of the regular polygons have the advantage of state transparency: the turtle returns to its original position and direction. When positioning turns are added, state transparency should be maintained by re-turning the turtle. For instance, with turns of 45, the procedures SQUARE.RIGHT and SQUARE.LEFT can be combined to produce DIAMONDS. TRIANGLE can also be turned to look like a BOWTIE. RECTANGLES rotate to look like wings.

Such ideas for designs are limitless. Abstract designs look like woven blankets. Butterflies and bugs are typical symmetry projects with combinations of all the shapes. I've been working on a leaf project with symmetry. Ideally, the shapes of the leaves and the veins are symmetric. However, children will find that real leaves are not precisely symmetric. People shapes and faces are other projects with many possibilities.

Combining shapes for a design points up the need for variable procedures. SQUARES or DIAMONDS are easier to manipulate if they have variable inputs.

AWAY FROM THE FOLD LINE

Limitations of drawing on the fold line become apparent as students create new projects. Eyes hooked together at the fold line make a funny face. Moving eyes away from the fold line produces more normal looking faces.

The mirror and pencil can be used once more to explore what happens as a line is drawn out and retraced back to the starting point. For a straight line perpendicular to the fold line, the turtle turns RIGHT 90, moves FORWARD, moves BACK, and turns LEFT 90. The mirror turtle makes opposite turns.

The forward and back motion may have been used in designs before to fill in circles or to draw other shapes. Two subprocedures could be created to take the turtle OUT.RIGHT and BACK.RIGHT. Then a subprocedure OUT.AND.BACK.RIGHT calls both of them.

```
TO OUT.RIGHT :ANGLE :DISTANCE
     RIGHT :ANGLE
     FORWARD :DISTANCE
     END

TO BACK.RIGHT :ANGLE :DISTANCE
     BACK :DISTANCE
     LEFT :ANGLE
     END

TO OUT.AND.BACK.RIGHT :ANGLE :DISTANCE
     OUT.RIGHT :ANGLE :DISTANCE
     BACK.RIGHT :ANGLE :DISTANCE
     END
```

The mirror procedures for this are OUT.LEFT, BACK.LEFT, and OUT.AND.BACK.LEFT. A master procedure OUT.AND.BACK pairs the symmetric lines.

```
TO OUT.AND.BACK :ANGLE :DISTANCE
     OUT.AND.BACK.RIGHT :ANGLE :DISTANCE
     OUT.AND.BACK.LEFT :ANGLE :DISTANCE
     END
```

The separate OUT and BACK procedures on either side are useful with PENUP, PENDOWN, and PENCOLOR to draw isolated shapes or shapes at the end of spokes.

```
     PENCOLOR 3 OUT.RIGHT 30 30
     PENCOLOR 5 SQUARE.RIGHT
     PU BACK.RIGHT 30 30 PD
```

The above commands place a square 30 steps away from the fold line at an angle of 30 degrees.

more Tipps

NOT THE STRAIGHT AWAY

Expand the notion of drawing to the left and right symmetrically with ZIGs and ZAGs. Walk and talk through a zig procedure.

"Move forward, turn right,
move forward, turn left."

After a ZIG.RIGHT is drawn, the reverse ZAG.RIGHT is needed to bring the turtle back to the starting position. The most common problem with this step is failing to reverse each movement from the end of the zig description. In this case, the zag would be:

"Turn right, move back,
turn left, move back."

When the zig and zag on the right side are completed, develop ZIG.LEFT and ZAG.LEFT with the same size but the opposite order of direction moves from those of the right side procedures.

Next, symmetric curves can be developed. The lessons from the zigzags are important for creating the reverse actions and the changes in directions for curves on either side of the fold line. For instance, a CURVE.RIGHT with FORWARD and RIGHT would require a CURVE.RIGHT.BACK which uses BACK and LEFT. Curves are good for antennae on insects, eyebrows on faces, mustaches, plumes, and the like.

ALL SIDES OF SYMMETRY

Building symmetrical pictures and designs with Logo demands problem solving with distance and direction in a way which is more involved than simple drawings. The turtle's insistence on being the center of the universe means that the programmer has to constantly adjust his/her commands to obtain the desired symmetrical results.

Projects begin with simple shapes on either side of the fold line. Only the first four primitives are needed. Procedures ease the exploration of more complex designs and possibilities. Input variables to procedures increase the range and speed of symmetrical drawings.

Establishing symmetry as an issue for discovery with Logo opens many new projects which demand thinking and creativity. More symmetry projects are suggested by Dale Burnett in Logo: An Introduction by Creative Computing Press (Ed Note: see review, NLX Vol 1 # 7 March 1983). Rotational and translational symmetry are included in addition to reflective or mirror symmetry.

Logo versions which have multiple turtles (Atari, Commodore, TRS-80 Color Logo, etc.) offer other methods of showing symmetry. With one turtle, each side must be drawn separately - one after the other. With multiple turtles, lines on the left can be drawn at the same time as those on the right. (Actually, the turtles take turns drawing on either side so that the two squares or two circles appear to be drawn simultaneously.)

Advanced students may want to create a procedure which will accept an instruction or a procedure for drawing on the right hand side and use list operations to transform it into the corresponding instruction or procedure for the left hand side and execute it!

Students may draw on paper or save (SAVEPICT) one side of a symmetric picture as challenges for each other. Drawing on both sides of a symmetric drawing means children will use all sides of Logo. It gives them the opportunity to develop their programming skills and problem solving strategies, and to produce some beautiful artistic work as well. They probably will have a little fun, too!

Steve Tipps is a professor in the University of Virginia's School of Education, and presents Logo workshops for school systems throughout the eastern United States.

The Computing Teacher Announces Logo Issue

The Computing Teacher will be publishing another Logo issue in December, as a followup to their extremely well received November 1982 Logo issue. To reserve your copy now, write to Sharon Franklin, ICCE TCT, University of Oregon, 1787 Agate Street, Eugene OR 97403, or call (503) 686-4414. Better yet, why not start a subscription to this outstanding journal?

MICROWORLDS

by
Glen Bull

Logo Grade Book
Part I Grade Averager

Last spring, I discussed some ways of teaching a computer to form plurals through list processing (Ed. note: see NLX Vol 1 #8 & #9, 1983). These techniques can be used to teach the turtle to average grades as well. In this column and the next, I will discuss methods of using Logo to do this.

CHARACTERLISTICS

Before continuing, it may be desirable to review some of the characteristics of lists in Logo. In Logo, a list is anything enclosed between brackets. A list of words can be printed.

```
PRINT [ THIS IS A LIST ]
THIS IS A LIST
```

A list may consist of just one word:

```
PRINT [ THIS ]
THIS
```

or even no words at all!

```
PRINT [ ]
```

A list consisting of brackets with nothing in between them is called an empty list. Among other things, an empty list can be used to print a blank line.

LISTS OF NUMBERS

A list can consist of numbers as well as words.

```
MAKE "JOHN'S.SCORES [ 89 94 83 91 ]
PRINT :JOHN'S.SCORES
89 94 83 91
```

It is possible to print either the first item in a list,

```
PRINT FIRST :JOHN'S.SCORES
89
```

or to print everything but the first item in a list.

```
PRINT BUTFIRST :JOHN'S.SCORES
94 83 91
```

LIST OPERATORS

These two list operators (FIRST and BUTFIRST) can be used to gain access to each of the scores in the list, one after the other.

```
TO EXAMINE :SCORES
     IF :SCORES = [ ] STOP
     PRINT FIRST :SCORES
     EXAMINE BUTFIRST :SCORES
     END
```

I was once told that, in England, the telephone is answered with the greeting of "Are you there?" The supposed reason for this is that if no one is there, there is no point in going on. I described this custom in class for a number of years until I was informed that people in England actually do not answer the phone in this way. (Would any of our British Logo Users Group colleagues care to respond?)

I was disappointed, but took solace in the existence of such a custom in Logo. The first line of the procedure EXAMINE asks if the list of scores is empty. If it is empty, of course, there is no point in going on.

If the list of scores is not empty, the first score in the list is printed. In the final line of the procedure, a new list is created, consisting of all but the first score in the list. In this way a series of smaller and smaller lists is created until, at last, the list is empty. Use of the procedure with John's scores produces the following result:

```
EXAMINE :JOHN'S.SCORES
89
94
83
91
```

TRACING THE STEPS

The process in the EXAMINE function is found in so many different uses of lists that it warrants examination in a step-by-step fashion. If you are not familiar with recursion, I invite you to study the process detailed below. In this illustration, the numbers which actually would be used by Logo are shown in place of the variable :JOHN'S.SCORES.

```
TRACE
EXAMINE :JOHN'S.SCORES
EXECUTING EXAMINE [ 89 94 83 91 ]
IF [ 89 94 83 91 ] = [ ] STOP
PRINT FIRST [ 89 94 83 91 ]
89
EXAMINE BUTFIRST [ 89 94 83 91 ]

EXECUTING EXAMINE [ 94 83 91 ]
IF [ 94 83 91 ] = [ ] STOP
PRINT FIRST [ 94 83 91 ]
94
EXAMINE BUTFIRST [ 94 83 91 ]

EXECUTING EXAMINE [ 83 91 ]
IF [ 83 91 ] = [ ] STOP
PRINT FIRST [ 83 91 ]
83
EXAMINE BUTFIRST [ 83 91 ]

EXECUTING EXAMINE [ 91 ]
IF [ 91 ] = [ ] STOP
PRINT FIRST [ 91 ]
91
EXAMINE BUTFIRST [ 91 ]

EXECUTING EXAMINE [ ]
IF [ ] = [ ] STOP
ENDING EXAMINE
```

SUMMING A LIST

The EXAMINE procedure does not do much, other than allow us to gain access to each score in a list. A slight change will allow us to add the scores and move a step closer to a grade book. To sum the list of scores, the first score in the list is added to a total, initially set to zero.

When all the scores in the list are exhausted, OUTPUT is used to pass the total to an operator, such as PRINT. The parentheses in the last line are not required, but are included in the listing for clarity.

```
TO SUM :SCORES :TOTAL
     IF :SCORES = [ ] OUTPUT :TOTAL
     OUTPUT SUM (BUTFIRST :SCORES) (:TOTAL +
FIRST :SCORES)
     END
```

Note that the initial zero state of :TOTAL is set by the input to the procedure.

```
PRINT SUM :JOHN'S.SCORES 0
357
```

AVERAGING A LIST

The average of the scores consists of the summed total divided by the number of scores. Another input, :NUMBER, can be used to count the number of scores. At this point, there are now three inputs to the procedure:

- the list of scores
- the summed total of the added scores (initially zero)
- the number of scores in the list (initially zero)

A master procedure which supplies the second and third items minimizes the number of inputs necessary each time a student's scores are averaged.

```
TO AVERAGE :SCORES
     OUTPUT AVERAGE.LOOP :SCORES 0 0
     END

TO AVERAGE.LOOP :SCORES :TOTAL :NUMBER
     IF :SCORES = [ ] OUTPUT (:TOTAL /
:NUMBER)
     OUTPUT AVERAGE.LOOP (BUTFIRST :SCORES)
(:TOTAL + FIRST :SCORES) (:NUMBER + 1)
     END

PRINT AVERAGE :JOHN'S.SCORES
89.25
```

ADDING NEW SCORES

There are two other functions which must be added before a usable gradebook is available:

- the ability to modify grades or add new ones
- the ability to save grades

A listing of global variable names (those created through use of a MAKE statement) can be viewed by the use of PRINTOUT NAMES (PONS in Apple (LCSI) Logo).

```
PRINTOUT NAMES
"JOHN'S.SCORES IS [ 89 94 83 91 ]
```

These variable names can be edited just as a procedure can, by typing EDIT NAMES (EDNS in Apple (LCSI) Logo). This places Logo in the edit mode. Instead of displaying a procedure in the editor, however, the variable names and corresponding values are displayed instead. After using the editor in the usual way to add new scores or modify existing ones, type CONTROL C to define the new names and return to the text screen.

more Microworlds

When the workspace is saved under a file name, both the procedures and variable names are saved. Although this will make it possible to save both the averaging procedure and the student's grades in the same file, it is esthetically more pleasing to me to save the averaging procedure in one file and students' grades in a separate one. The following sequence is one way this could be accomplished.

- Type in the commands for the procedures AVERAGE and AVERAGE.LOOP

- Verify that they work properly, using a set of trial grades, such as "JOHN'S.SCORES

- Erase all names in the workspace with ERASE NAMES (ERNS in Apple (LCSI) Logo), then save the two procedures in a file name such as "GRADE.AVERAGER

- Clear the Logo workspace by typing GOODBYE (ERALL in Apple (LCSI) Logo)

- Type in the grades for all the students in the class:

```
MAKE "JOHN'S.SCORES [ 89 94 83S 91 ]
MAKE "SALLY'S.SCORES [ 92 91 89 94 ]
etc.
```

- Use the editor to correct any mistyped grades. This can be done by typing EDIT NAMES (EDNS in APPLE (LCSI) Logo) and making alterations just as you would in a procedure. Type CONTROL C to define the new changes.

- Save the grades in a file, such as SAVE "ENGLISH.GRADES

Placing the grade averager and the class grades in separate files makes it convenient to create files of grades for other classes as well. To use the averager, load the file containing the averager into the workspace, and then load the file containing the grades of the class you wish to average.

It is seldom the case that all students are present for every examination. It might be desirable to indicate which students were absent from an examination by including a note such as ABS in the list of scores. For example, assume that John missed the third examination. In that instance, the values for "JOHN'S.SCORES might be recorded as [89 94 ABS 91].

If this method of recording missed examinations is used, a slight change to AVERAGE.LOOP is necessary. Otherwise, the procedure will attempt to add ABS to the total of the other scores: 173 + ABS = ?

Since letters cannot be "added" to numbers, Logo will complain:

```
+ DOESN'T LIKE ABS AS INPUT
```

This potential problem can be removed by including a test to determine whether an item is a number before adding it to the total of the other scores.

```
TO AVERAGE.LOOP :SCORES :TOTAL :NUMBER
     IF :SCORES = [ ] OUTPUT (:TOTAL /
:NUMBER)
     TEST NUMBER? FIRST :SCORES
     IFFALSE OUTPUT AVERAGE.LOOP (BUTFIRST
:SCORES) :TOTAL :NUMBER
     OUTPUT AVERAGE.LOOP (BUTFIRST :SCORES)
(:TOTAL + FIRST :SCORES) ( :NUMBER + 1)
     END
```

LCSI version of procedures

```
TO AVERAGE :SCORES
     OUTPUT AVERAGE.LOOP :SCORES 0 0
     END

TO AVERAGE.LOOP :SCORES :TOTAL :NUMBER
     IF EMPTYP :SCORES [ OUTPUT :TOTAL /
:NUMBER ]
     TEST NUMBERP FIRST :SCORES
     IFFALSE [ OUTPUT AVERAGE.LOOP (BUTFIRST
:SCORES) :TOTAL :NUMBER ]
     OUTPUT AVERAGE.LOOP (BUTFIRST :SCORES)
(:TOTAL + FIRST :SCORES) (:NUMBER + 1)
     END
```

Glen Bull is a professor in University of Virginia's School of Education, and teaches Logo courses at both the graduate and undergraduate level.

Logo Bulletin Board Test

The National Logo Exchange and the University of Virginia are conducting a pilot study of an electronic Logo bulletin board system. We are seeking volunteers to assist with the field test. If you have access to a computer with a modem and would like to help, please send your name and address to NLX LOGO BULLETIN BOARD, PO Box 5341, Charlottesville VA 22905.

Commodore Logo: Let's Take a Look

by

Steve Tipps

Commodore Logo, developed by Terrapin for the Commodore 64, is a giant turtle step in the right direction for Logo users.

Imagine all the features of Terrapin's Apple implementation. Then, add 7 programmable sprites which take advantage of the Commodore's highly regarded graphic capabilities. Next, include a handful of commands for easier list processing and additional graphics commands. Finally, consider 30% more working memory available. All of this is packaged with a good Logo tutorial and reference manual.

This is a Logo version that a skinflint could love. The Commodore 64 system with CPU-Keyboard, color monitor, and 1541 disk drive retails for $750 and can be found on sale or on school contract for less. The Commodore Logo lists for $60, but has been on sale in the Washington DC area for $40 to $50. A full fledged Logo system for under $800 makes one want to run right out and buy. As resident skinflint at NLX, I did just that after a week of playing with a Commodore system loaned to me to develop an inservice for Falls Church VA Schools.

COMMODORE 64, LOGO AND ME

The transition to the Commodore 64 keyboard was easy, with CTRL being the irritant. Since Commodore's CTRL is located up a row from that of the Apple, my left little finger hits the RUN/STOP key. The mistake led to a happy discovery that the RUN/STOP key also defines a procedure and to the unhappy event that I "define" something inadvertently when I'm trying to use editing keys. My fingers will learn. I like the function keys being programmed to change TEXTSCREEN to SPLITSCREEN to FULLSCREEN.

Loading Logo requires typing instructions to Commodore's operating system. Loading is slow and noisy. Since I hate PLEASE WAITing, I boot up while getting organized to work.

All the old favorite turtle graphics commands are available with a new palette of 16 colors. Numbering of the colors is somewhat confusing because the colors written on the number keys do not work for Logo. The Commodore 64 uses a numbering system for colors 1 to 16, while Logo wisely uses 0 to 15. The "1" key is marked "black," but draws white in Logo. The "2" key is marked "white" and draws red, and so on. To correct the discrepancy between the labels and the Logo colors, I suggest that stickers be colored and taped over the printed names on the front of the numbers keys.

A new drawing command is DOUBLECOLOR. This draws a wider line. DRAWSTATE reports nine pieces of information about the state of the screen, such as pen color, background color, screen in use, pen state, and the like.

Here's a change Terrapin users will appreciate. CTRL P retypes automatically the last line entered, even if it includes a REPEAT command!

To me, the most exciting new graphics feature is stamping letters on the graphics screen. STAMPCHAR allows typing of single letters on the screen. Students often want to write on the graphics screen or label their drawings. My first major programming task with Commodore Logo involved procedures to write entire words and sentences on the graphics screen rather than one letter at a time. My sense of accomplishment was not diminished by the discovery that Terrapin had kindly provided similar procedures on the utilities diskette. In fact, I like mine better than theirs! My next Logo task is to attempt graphics from the KoalaPad with labeling.

Sprites on the Commodore are on the utility diskette. Assorted sprite shapes, such as animals, vehicles, geometric forms and an animated runner, can be read from the diskette. The demonstration program only hints at the possibilities for manipulating sprites. Many more examples of what to do with sprites will be needed.

I believe that these multiple drawing sprites will radically change what can be done in graphics. I am rethinking some of my old projects with the new possibility of having (nearly) simultaneous events. The multiple sprites are new objects to think with. These new capabilities enlarge the range of possibilities.

Commodore continued

BIGX and BIGY with SMALLX and SMALLY allow changing the scale for the sprites so that the kangaroo can be tall and thin or wide and short. The shape grid for creating the sprites is similar to other grids in TI Logo or Pilot -- all of which I find frustrating. The triangle (sprite 0) can be turned into a real turtle shape or any other shape which is desirable. I am inspired by the idea of Multiple turtle Instant Logo.

The sprite demonstration game TRAP shows another new feature of line detection. The turtle detects when it touches a line drawn on the screen. Line detection should lead to many new projects in Logo. The game applications are most apparent, but applications such as science experiment simulations and map making should be developed also.

A major improvement has been inclusion of list primitives COUNT, EMPTY?, MEMBER? and ITEM. Some philosophical argument might be made that people need to create these procedures. However, my experience is that students need and can use these procedures before they have the sophistication to write the programs for them. I prefer them as primitives because they allow more immediate entry into interesting list projects.

The written documentation by Virginia Carter Grammer, E. Paul Goldenberg, and Leigh Klotz is comprehensive. It includes separate sections on graphics, computation, words and lists, sprites, and music. Various graphics projects and list projects are also included. My main use ot the manual has been the excellent index, appendices and reference section at the end. I rate it much better than average for written documentation. The utilities disk has the usual games and demonstrations and also includes useful routines for printing and plotting on the Commodore peripherals.

THE ADMIRABLE COMMODORE

My admiration for Commodore Logo and the Commodore 64 has continued to grow in the six weeks I have been working with them. I have no serious complaints and quite a few compliments.

My positive reaction to Commodore Logo was supported by the local Commodore 64 fan, John Strang, with only a few specific quibbles. He doesn't like the manual organization as well as I do. He wishes sprites and music were built in rather than on the utility disk. John, an eighth grade student, is pleased overall and has found the line detection excellent for his current project, a treasure hunt maze.

Testimonials are always suspect in computer land, but I put money on the line for this one. After waiting two years, I found a system which had enough power to do what Logo should do and was priced in the range I could afford personally.

The single biggest improvement for me is the luxury of increased working memory. The dreaded OUT OF STORAGE, PLEASE ERASE SOMETHING message has not appeared, even after two or three hours of programming without attention to cleaning up my workspace. Terrapin seems to have packed more for less in Commodore Logo. Congratulations on an excellent job.

Conference Notes

MIT's Laboratory for Computer Science has initiated plans for a series of National Logo Conferences to be held periodically, beginning in 1985. To be placed on the mailing list for information on the 1985 conference, send your name and address to 1985 Logo Conference, Attn: Jacqueline Karaaslanian, MIT Laboratory for Computer Science, 545 Technology Square, Cambridge MA 02139.

A limited conference will be convened in Boston, June 25-29, 1984, to give those most active in teaching and developing Logo culture the opportunity to interact and to construct the framework for the 1985 gathering. If you have ideas or suggestions which you feel should be incorporated into these plans, please send them to the address above.

The 6th annual National Educational Computing Conference (NECC) will be held in Dayton, Ohio, June 13-15, 1983. A call for participation has been issued for papers, presentations, and workshops. If you have been looking for a national forum for reporting your Logo work, here it is. For more information, write Lawrence A. Jehn, Department of Computer Science, University of Dayton, Dayton OH 45469.

NLXual Challenges with Robs Muir

The debate about the "best" computer language for education will undoubtedly rage for years to come. What makes one language better for education than another? The polemists disputing the relative merits of APL, BASIC, Pascal, Logo, Lisp, and the like seem to have lost sight of the principal benefit of computational learning environments, that of practice in problem solving. It is hard to see the forest for the tree structures.

Although the descriptors may vary, there has been a fair amount of academic research into the nature of problem solving. The subject of heuristic thinking (or mathetics) has been examined repeatedly throughout history. Scholars such as Euclid, Pappus, Descartes, Leibnitz, Polya, and Bolzano have done much to articulate the methods and rules of invention, improvisation, and discovery. Yet, much remains to be discovered.

Academic research is one thing; the practical application is another matter. Problem solving techniques have had little opportunity to be included in the classroom curriculum except as an incidental byproduct of the traditional course of study. However, a number of teachers study the process of learning and successfully encourage their students to examine and refine their own heuristic techniques.

The importance of a computer in education may ultimately lie in its ability to stimulate problems of immediate importance to the student. Thus, the importance of a computer language lies in its ability to reflect a student's thinking.

Problems encourage active learners to wrestle with new knowledge, to invent solutions. In the course of discovery, a special excitement is generated that stimulates further inquiry. This heady experience is infectious. It can stimulate an entire group to "discover" great things which will not be easily forgotten.

But practice with problem solving must not stop outside the classroom walls, nor should it cease after one acquires a diploma (or a teaching certificate!). This must be so, if we truly believe in "life long learning." Indeed, teachers should be the group most aware of heuristic techniques.

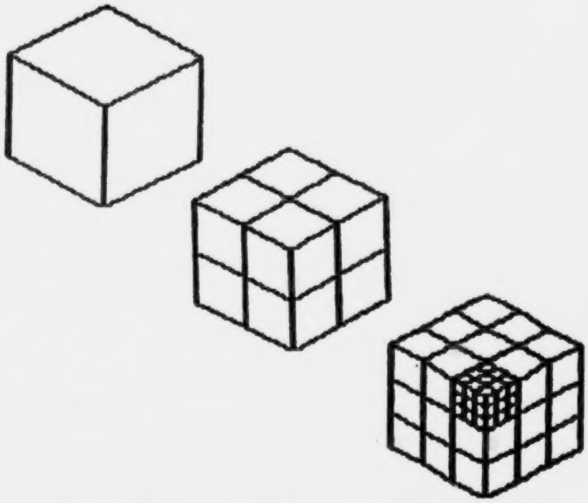
With this in mind, the NLX is beginning a series of articles devoted to Logo problem solving challenges. Each month, a problem situation will be presented, along with a suitable intellectual challenge. Your goal is to arrive at a personally satisfying solution.

There is no single solution to these puzzles; in fact, there may be a great many possibilities. We trust that, by striving for your solution, you will come to better understand the Logo philosophy and the Logo family of languages. More importantly, we hope that you will come to better understand your own unique approach to problem solving.

This month's problem grew from a desire on the part of Frank Nakase, a teacher in Huntington Beach CA, to familiarize his students with two dimensional representations of three dimensional objects.

This cube problem is a good exercise in procedure building. The goal here is to create a cubelike drawing that is capable of being extended to deep levels of complexity. As with most Logo programming (and problem solving in general), look for the simple case, find instances of repetition, and build from there.

Here are a few examples of some typical drawing developments.



Depending on the approach you take to this problem, you may discover some interesting things about the relationships of the sides (or cubies, as they sometimes are called). For example, is the simple case a hexagon with three radial lines, or three parallelograms rotated?

Challenges continued

If you are interested in exploring the mathematical ideas embodied in a cube, Chapter 6 of Abelson and diSessa's Turtle Geometry (see review, NLX Vol 1 #2 October 1982) contains an innovative treatment.

✓ Note: As with all NLXual Challenges, if you want to receive a possible solution, send a copy of your solution (or your attempt) with a legal size self addressed stamped envelope to NLXual Challenges, Attn: Robs Muir, 1688 Denver Avenue, Claremont CA 91711. We will compile contributions and send you the most elegant solution.

If you would like to submit a challenge or puzzle for this column, send your ideas to the above address.

Robs Muir is a teacher in the Claremont CA Unified School District and Claremont Graduate School. He presents Logo workshops in the southern California area.

Help Wanted for Logo Book

Here is an opportunity for you to contribute to a Logo book! Ray Hammond, a member of the British Logo Users' Group, is gathering material for a book entitled, "Logo and Your Child." He is interested in collecting case histories of school Logo experiences, reports, and articles, and can reimburse reasonable expenses for duplication and shipping, if arranged in advance. Write to: Ray Hammond, Tivoli House, Greenway Lane, Bath BA2 4LL, England, for more information.

News from LCSI

Logo Computer Systems Inc. (LCSI) has announced the availability of several aids for Apple Logo users.

Two public domain diskettes, the Apple Logo Sample Program Disk and the Apple Logo Tool Kit, are both available for copying from most local Apple dealers. The reverse side of the Sample Program Disk contains extensive documentation in files compatible with Apple Writer II or IIe. Diskettes may also be ordered direct from LCSI, for $12.95 US or $14.95 Canadian.

A 64 page manual containing the printout of the diskette documentation may be ordered for $3.25 US or $3.75 Canadian.

The Apple Logo Presentation Manual contains materials sufficient for organizing Apple Logo presentations from one-half hour overviews to five-day workshops. Included are the areas of Logo introduction, developing problem solving skills, and use in the content areas. The cost is $29.95 US or $34.95 Canadian.

LCSI also has organized a series of five "on the road" Logo courses which can be taught at your location. Included are Introduction to Logo, Teaching Logo, Advanced Logo, Inside Logo, and French Logo. School system consulting services are also offered.

LCSI addresses and telephone numbers are 220 Fifth Avenue, Suite 1604, New York NY 10001, (212) 684-0710, or 9960 Cote de Liesse, Lachine, Quebec H8T 1A1, (514) 631-7081.

The National Logo Exchange, copyright 1983 by Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly, September through May, $25 (US) per year, mailed FIRST CLASS from Charlottesville, Va. $5 (US) additional per year for addresses outside the USA, Canada, and Mexico. The opinions expressed by the authors are not necessarily those of The National Logo Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress Street, Salem, MA 01970. Address other correspondence to: The National Logo Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor....................................Tom Lough

ISSN 0734-1717

The National LOGO Exchange

Volume 2 Number 4 FORWARD 100! December 1983

Students Write The Book On Logo

by Lynne Mass

As I read Nancy Sopp's article, "Logo Messups: Process vs. Product," in the October 1983 NLX, I nodded my head in agreement and chuckled, remembering last year when my students and I also proposed to learn Apple Logo together. We began easily, but then reached frustration with the tutorial and decided that we needed an additional manual. Since we could not find one to suit our needs, we decided that maybe the world needed another Logo book, written for children. And why shouldn't *we* be the ones to write it?

How does a teacher begin to write a computer book with ten year old co-authors? Let me digress a bit.

When our school got its first computer and I taught myself the initial BASIC lessons, I wrote them down for my elementary age students to use in my classroom so they could begin to learn programming, too. They gobbled up the lessons almost as fast as I could produce them! Then came the dreaded day when I knew they were ahead of me ... I couldn't answer their questions!

Having been left in the "computer dust" by my eager students, I decided to work on converting my manual to publishable form as more and more lessons evolved. While I slowly gained some programming knowledge, I frequently reached levels of unrelenting frustration. One day, in utter despair, I bellowed right out loud, "I am so dumb! I can't figure this out!"

Debbie, a fifth grader with a twinkle in her eye, mercifully broke the silence that followed with, "You're not dumb. You're just a slow learner!" We all had a good laugh on that one!

Then Apple Logo arrived! We began investigating this new world, still vaguely discussing the possibility of publishing our own manual. It was not until my BASIC manual was printed that the children really got excited about writing! They organized a format for their manual, decided what to include, and proceed to teach themselves the essence of turtle graphics. After agreeing upon a title, the students developed each lesson in rough form. After rewriting, it was "field tested" in class, and further revised.

After helping to bring it to final form, I contacted Tom Kemnitz of Trillium Press in New York. He sent the students and me contracts to sign, with royalty agreements. It was fun to sign them with the children (and their parents).

Our writing project developed Logo skills, writing and editing skills, and introduced the students to the real world of publishing, publisher's requirements, deadlines, compromises, and legalities. When I asked them what they liked the most, they shouted, "The royalties!"

Whatever it was that motivated them, they were so enthusiastic that they began their second manual on TI Logo ... when summer vacation began! This has just been completed, and will appear in print soon.

As happened to Nancy Sopp, my students and I learned alongside each other, working as colleagues. We learned about Logo and publishing and having fun at the same time!

(Ed. Note: Kids Working with Computers: An Apple Logo Manual for Turtle Graphics is available from Trillium Press, Box 921, Madison Square Station, New York NY 10159.)

Lynne Mass is an elementary science coordinator at Abington Friends School, Jenkintown, PA, and has published a series of books entitled, Kids Working with Computers.

Guest Editorial

by Steve Tipps

Pushing the Turtle

Glen Bull and I wrote an article for the special Logo issue of The Computing Teacher (December/January). The article, entitled "Problem Spaces in a Project Oriented Logo Environment," is also printed in serial form in this issue of the NLX and the next. The article is the result of two years of trial and error with Logo. Logo is as exciting as it was two years ago, but our ideas about using Logo are not the same. We have grown and changed in our understanding of Logo.

We were very idealistic in our beginning days, because we felt the benefits of Logo were self evident. Logo was a model for what we believed education should be - a continual process of discovery and collabortive learning. Logo has borne out our beliefs. However, two years have forced us to be more realistic about the world Logo lives in.

While Logo's world is infinitely extensible, our world is not. Logo knows no boundaries of time; we do. Logo is patient and good natured eternally, demanding only that we obey the rules which we give it. We are neither eternally patient, nor infinitely good natured, and only rarely consistent about the rules we like and want. As brilliant as Logo is as a model for learning, it lives in an imperfect human environment.

In our article, we have tried to communicate our view of how Logo might coexist with mortals. The first time we suggested that a project with Logo might be the creation of a microworld, our students (classroom teachers) looked at us as if we were crazed. They retorted that they could not make a DYNATURTLE world, and demanded to know what other microworlds there were. It was the only example we had. We had a bug in our communication and our thought.

The idea of a problem scheme - designed by a teacher for guided discovery and laden with implicit information - has been a thought of Glen's for two years. Playing with NIM in one class convinced us that Logo could be used for programming, problem solving, and learning about number theory. But, other than ANIMALS, we could not find examples which met our criteria for well developed problem schemes. Assignments to invent problem schemes fell on confused ears.

Next, we suggested Logo projects which were useful personally or professionally. We wanted people to see the potential of Logo to do anything that they could dream up. Challenge the turtle to be used in everyday life, not just to draw squares. This assignment resulted in exciting applications: audiogram plotters, report forms for reading and hearing tests, sports scorekeepers, various INSTANT arrangements, and many cooperative classroom projects. We were beginning to get the hang of describing Logo projects.

"Problem Spaces" is our most recent attempt to suggest Logo projects. Many elements have been organized into four problem spaces which describe different kinds of Logo learning. If the plan does not communicate immediately, we ask that you reread it six months from now. If our framework inspires you or incites you, please let us know. The ideas are worth debating. Debugging the problem spaces seems essential in broadening the niche for Logo. Logo needs more room to survive in our human world.

By looking at Logo in a bigger way, more opportunities for personalizing learning will be available to teachers and students. We want both teachers and students to see their "subjects" in a new way. With Logo, information becomes the byproduct of learning, rather than the end product. The end product of learning should be the desire and the need for more learning.

Finding good Logo examples is still a problem, but we are working on it. We are convinced that books with 2001 preprogrammed Logo projects are not the answer. Good examples are likely to be the result of groups of teachers and students pushing the turtle into new areas. The aim of the NLX is to provide the Logo community with a forum for sharing ideas that work or nearly work. Reading the NLX is half of your involvement in the Logo task. Your ideas and questions are needed for the NLX to fulfill its mission. Consider the NLX an invitation.

FD 100!

Steve

Problem Spaces in a Project-Oriented Logo Environment

by Glen Bull and Steve Tipps

(Ed. Note: This article, published here in two parts, appears in its entirety in the December/January issue of The Computing Teacher. The article may be reprinted provided full credit is given the authors and the NLX, and that a copy of the reprint is sent to the NLX.)

Part I

A recent review in Consumer Reports characterizes Logo as a language that has gained a wide following among educators as a way to teach programming concepts to children before they have sufficient reading, writing, and math skills to tackle the BASIC language. (See "The Computer as Teacher," Consumer Reports, Vol. 48 (November 1983), p. 617.)

Programming concepts are readily learned through Logo, but Logo is more than an efficient way to introduce programming. Such misunderstandings about Logo are not uncommon. Many introductory texts discuss Logo solely as a "language for children" or as a "way to teach computer literacy." These statements unduly restrict Logo's intent, both as a language and as an educational philosophy.

Using Logo also develops problem-solving skills; the structure of the language encourages a modular approach to problems. Features such as a resident screen editor facilitate a strategy of successive approximations to a solution, since procedures can be altered in seconds.

Logo can also be a vehicle for learning content. Papert presents the imagery of learning math in "Mathland" in the same way that it is possible to learn French through a stay in France. Acquiring programming and problem-solving skills, and developing concepts in subjects such as math and English are important parts of Logo environments.

Of these three, the potential of Logo as a meaningful way of learning content has been neglected. Using Logo for learning subjects ranging from beginning reading to college physics have been imagined, but such applications, though being explored, are not yet in widespread use. Classroom activities which use Logo to explore and develop concepts in subject areas are needed as alternatives to traditional didactic teaching styles. Otherwise, the Logo environment is incomplete.

PROJECT ORIENTED LEARNING

Consider the following proposition:

"Effective Logo learning is project oriented."

In a project, the content to be learned is embedded in a goal meaningful to the learner. The need for personally satisfying goals can be found in initial use of the REPEAT statement. The first need for repetition can be induced by a desire to avoid repeated retyping of an instruction, but children soon discover great joy in the unpredictable effects of randomly typed commands repeated a large number of times. These seemingly random instructions often produce beautiful and symmetrical designs. Some exploration of this type must be encouraged. However, failure to go on to purposeful activity will produce a row of children gazing in stupefied boredom at 5,000 shifts in the background color of the screen.

To return to Papert's imagery, French is most effectively learned when it is embedded in another context and is not in itself the primary goal. Yet even the presence of the French environment is not enough - Americans in Paris have survived on hamburgers from American hamburger chains.

Subjects such as mathematics and English are also learned best when embedded in a meaningful project. To the student using Logo, the subject matter content learned should simply be something which must be understood in order to do something else which the student wants to accomplish.

Finding projects and applications which address specific content areas seems to be a prime difficulty in implementing Logo based school programs. The problem stems from perceptions of limitations and insufficient illustrations of classroom use rather than

Problem Spaces continued

the language itself (any memory limitations aside). The relative newness of Logo on widely available microcomputers means that most teachers and students have not pushed the boundaries of Logo beyond turtle graphics. The notion that teacher selection of specific content invalidates the discovery principle is also an inhibiting factor.

Removing these barriers permits a broader view of Logo and makes opportunities for learning content areas with Logo more apparent.

PROBLEM SPACES

Two dimensions are important in defining how Logo might be used in schools:

- Who provides the idea for the project?
- Who solves the problem and programs the solution?

A two-dimensional matrix defines four possible problem spaces in a Logo environment:

Implementation (Solution)	Source of the Project	
	LEARNER	FACILITATOR
LEARNER	I	III
FACILITATOR	II	IV

I. Learner Generated / Learner Implemented Problem Spaces.

In a Type I project, the learner generates both the problem and the solution. This is often considered to be the most natural, "Logo-like" environment.

Even in Type I Problem Spaces, the teacher has considerable control over the content to be learned. If the teacher supplies the set of primitives dealing with turtle graphics, the learner is apt to develop projects which involve geometry, physics and trigonometry. If the teacher supplies primitives which can be used to take apart and combine words and sentences, the learner may embark upon poetry generation and other projects which involve a natural language.

If the teacher supplies tailor made procedures, the learner is apt to explore that particular set of ideas. A BOUNCE procedure modeling reflection of light rays or billiard balls would encourage exploration of vectors.

The underlying structure of Logo encourages strategies such as patterning, modularity, comparison and approximation. At the same time, the learner discovers the possibilities and limitations of the rules which govern the natural microworld of the turtle. This world offers opportunities for learning a new vocabulary, enhancing thinking skills and discovering a consistent rule set.

Ideally, each new project inspires further learning. Each bug, error, or "what if" demands satisfaction and solution. The need for REPEAT begins with pattern recognition. The rationale for procedures is developed when Logo users need an easy way to recall previous events and change them. Implementation of REPEAT statements or of the editor cannot be transmitted telepathically. Providing information about the vocabulary and structure of Logo is the facilitator's responsibility. A facilitator must at time instruct.

The natural learning space of exploration in Logo is not without difficulties for both learners and facilitators. This environment of exploration demands a non-authoritative role for teachers. When a learner asks, "Can I accomplish this?", the answer may be, "How do you think that might be done?" or "Show what you want the turtle to do." These replies are invitations for learners to extend their skills. Some people discover new applications for known skills as they describe their intended goal. Others seem to refuse the challenge of thinking about their own question. They are discouraged because Logo is not automatic ... like a video game.

II. Learner Generated / Teacher Facilitated Problem Spaces

In a Type II Logo project, the learner may generate an application which requires skills beyond those possessed. The teacher can choose to direct the learner to an area which is more consonant with that learner's existing level of computing skills. However, there can be value in working out the description (and, if possible, a solution) of the problem in English, even if someone else codes the program. A system analyst, for example, may work out the flowchart or pseudocode for a problem that is coded by a team of programmers. In a TYPE II project, the learner fills the role of system analyst, and the teacher serves as programmer by providing the actual coding.

Beyond the skills required for development of the logic underlying the solution, two other benefits may result from a Type II project. The learner may have the opportunity to observe firsthand the false starts and problems encountered as the teacher attempts to code the problem. When a learner asks how to do something which is not immediately apparent to the teacher, the question may become the basis for a cooperative project. This type of collaboration is not possible when the learner senses that the teacher already knows the answer.

The second benefit is the possible introduction of a new command or concept in a meaningful context. The teacher's version of the program can serve as a model or exemplar which the learner can use to generate solutions to similar problems.

(To be concluded in the January 1984 NLX.)

NLX ABC's Lead to List Processing Adventures

by Jim Fry

There is life beyond turtle graphics. I found this out while teaching Logo to a summer class of 8 students between the ages of 8 and 13. After a week of turtle graphics, we began using the NLX ABC's to create patterns. Soon, the students were drawing words and sentences on the graphics screen with them.

From there, they showed an interest in printing on the text screen. Thus began our adventures with list processing. After teaching Logo how to print out lists of names and telephone numbers, they began to search for more ambitious projects.

Why not an adventure game? The idea was an instant hit!

(For the benefit of NLX readers who are not familiar with adventure games, here is a brief explanation. A set of rooms or caves or locations is programmed into the computer, based on a map drawn on paper. Connections between the locations are specified so that players can move from one place to certain adjacent others. Events are programmed to happen when players reach certain locations. For example, a monster might attack, or a treasure might beckon. Sheer survival is one important objective of an adventure game!)

We decided to make our adventure take place in a house with four rooms and a main hall. Players could move from room to room by typing compass directions N, S, E, or W.

Here is a sample of what might happen to you. You start at the front door, with the option of entering or chickening out. When you enter, you find yourself in the main hall. Logo asks, "Which direction?" You type N, and find yourself in the ballroom. You spy a monster lurking in the dark corner! The computer asks if you want to fight or flee. If you fight, you must choose a weapon. If you select the right one, you kill the monster. But, if you pick the wrong one, you die, followed by, "Do you want a large or small funeral?"

There are many other surprises waiting in the adjoining rooms. My students had some exciting times planning, writing, and programming the game. They also enjoyed watching others play it. And, they learned a lot about list processing along the way! I was pleased to find the NLX ABC's acted as a natural transition to the list side of Logo.

(Ed. Note: For a free listing of the procedures used for this adventure game, send a self addressed stamped envelope to Jim Fry, 37279 Six Mile Road, Livonia MI 48152.)

Jim Fry teaches math, reading and Logo to elementary school students in Novi, MI, and is the coordinator of LIFT, an organization to help Logo teachers through workshops, consultations, and materials. He is a frequent contributor to the NLX.

TIPPS for TEACHERS

by
Steve Tipps

The Issue of Instant

Instant Logo, or single key Logo, is a set of procedures which moves the turtle around without typing the entire command. For instance, the F key might automatically cause the turtle to move FORWARD 10 and the B key BACK 10. Programs of this kind are available on utility disks, in many Logo publications, including Abelson's Logo for the Apple II and Apple Logo (Byte-McGraw Hill, 1982), and as commercial software such as Delta Drawing by Spinnaker.

Philosophical and mechanical issues of Instant need discussion. Should Instant be used at all? If so, why? When? How? What are the features that make an Instant program good or bad?

After playing with various Instant programs, I have concluded that Instant:

1. is not a substitute for Logo.
2. can serve as a bridge to Logo.
3. allows people access to the power of computers.
4. provides a model for making the keyboard "alive." This has interesting possibilities for many projects in Logo.

I will expand these propositions.

NO SUBSTITUTE

Instant is typically associated with introductory turtling. The turtle can move around the screen, draw figures, change colors, clear the screen, and even recall and replay its previous moves. This is a subset of beginning Logo features.

Logo is more than turtle graphics. Full Logo must exhibit capabilities in list and numerical operations, and possess a modular structure which encompasses recursion, variability, and interactivity. Finally, and perhaps foremost, Logo must be extensible, that is, it must have the capability to expand through the creation of new commands.

By definition, Instant is not Logo. In my opinion, advertisers' claims that a software package which moves a thing around a screen is "better or easier or cheaper" than Logo are nonsense. You might as well assert that looking at a postcard of the Eiffel Tower is better and easier and cheaper than going to Paris. The two experiences are not equivalent.

Could someone invent a single key Mega-Instant which included all of the features of Logo? Given enough keys on the keyboard for all Logo commands and enough memory to accommodate both the Instant program and the history being generated, a Mega-Instant is possible, but pointless. Instant puts a cocoon around Logo which may obscure the immediacy and flexibility which are part of natural Logo. However, the cocoon can serve a purpose. A well designed Instant serves as a stage in learning a little about Logo.

BRIDGE

A philosophical question is how to design an Instant so that unintended Logo learning is not lost. I propose a limited definition of Instant turtling as an introduction to Logo. No instant program should do for a child what that child is capable of doing independently. Too much Instant robs the child of the delight of working out problems for themselves.

Teachers in our Logo classes write essays in response to any idea they find interesting from Mindstorms. They write passionately, critically, and perceptively about Logo, teaching, and learning. Two passages by teachers at Thomas Jefferson Elementary School in Falls Church, Virginia, illustrate two approaches in learning to make a square.

Jane Quick wrote:

"Two of my third graders wanted to make a box. They gave the commands FD 100, RT 100, FD 100. At this point they realized that the turtle had turned more than the square corner they wanted. Although just a little uncertain what to do now, they went back and tried RT 80 (too small), then 89 (still too small), then 95 (a little too big), 92, 91,and 90. They knew immediately that they had it! They had discovered for themselves that a square corner was exactly 90 turtle steps.

"They were thrilled by their discovery! They had bolstered their self confidence. The next, perhaps more difficult, challenge would be less threatening for them. They had discovered that their initial mistake was not really so terrible, because it had been relatively easy for them to fix their thinking by working together."

On the other hand, Carol Monsess reports:

"I have several students this year who had taken a computer course in Logo this past summer. When they told me, I got the impression that they knew a lot about Logo. They certainly had learned a lot of terms; they could tell me the names of the pieces of equipment and also seemed to know how to make the turtle move and to make squares and various other shapes. I set up the computer for several of them to work on and to show me what they could do.

"After a few minutes, one came and told me there was something wrong with the turtle because it wouldn't make a square. So I went over and everything looked like it was hooked up right and working. I asked them to show me what they had done. The little girl pushed the letters SQ and said, 'See? When I push the SQ, the turtle didn't make a square!'

"This seemed to be a perfect example of the wrong way to teach Logo to children. Given this powerful learning tool and bright children, all the learning had been taken out of Logo. I explained how this turtle had to be told what to do in order to make a square ... Later, the same child told her mother that, if they got a computer, she wanted one that knew more because it was less work. It made me realize that ... even Logo can be taken and taught in such a way that much of the learning is lost."

Instant programs should not take the learning out, but should be a transition for children into Logo. The mechanics might be very different for children of different ages and abilities. The typical Instant turtling program uses keys on the keyboard as shorthand or abbreviations for actual Logo turtle commands.

F key for FORWARD
R key for RIGHT
C key for CLEARSCREEN
etc.

The turtle moves immediately upon touch of key. The programs are available in Harold Abelson's and Dan Watt's books and on Logo utility disks, and should be considered as models for your own Instant procedures rather than fixed programs to be used exactly as they are written. The Instant program in the book may not be the right one for your purposes.

DISREGARD QWERTY

For some children (ages 3 and 4, and physically limited), the goal of Instant is to remove the keyboard effectively as a barrier to turtling. Control of the turtle is the objective. However, the specific letters may not be important. Design an Instant program and disregard the letters on the keyboard. Keys can be covered with adhesive dots and made into anything you want: arrows, shapes, symbols. The Instant keyboard can be arranged more spatially, or logically than is dictated by the QWERTY layout.

For instance, the RIGHT turn could be controlled by a key on the right side of the keyboard and the LEFT turn being on the left side of the keyboard. Presently, the L is on the right side, and the R is on the left side. So much for teaching any meaningful spatial responses. Perhaps the R key could be used for LEFT turn and U key for RIGHT.

Choosing keys for FORWARD and BACK is complicated because they are easily confused with up and down. Putting them on the same row seems like the best arrangement. Perhaps you might use the F key for FORWARD (coincidentally) and the J key for BACK. My personal design goal is placing keys so as to encourage putting fingers on the home row of the keyboard.

The screen commands such as DRAW, HOME, and CLEARSCREEN might be added on the bottom row keys V, B, and N respectively. The basic Instant keys would be arranged in the center of the keyboard.

With those few Instant keys, turtling is possible. Embellishments are infinite. The teacher can respond and add commands as the children seem to grow to them. HIDETURTLE and SHOWTURTLE could be added at the center of the keyboard on H and G. Pen colors and background colors might be added on the top row of the keyboard by sharing the same number keys. Background colors could be reached using the shift key. Shapes such as squares, triangles, stars, and circles could also be programmed and made available through other keys.

The sizes of the turtle movements and shapes should be scaled so they can be used together. After experimenting with different sizes, I think that movements of 20 or 30 turtle steps and 30 turtle turns are a good starting place. Smaller sizes don't seem to have enough effect to be really interesting.

Whether you add colors or shapes or anything else is your choice. The first law of Instant is to suit the procedures to the level and needs of the children without denying them the joy of discovery. For a four year old, having an Instant square may inspire creative squaring. For a seven year old, an Instant square may interfere with finding RT 90. For a cerebral palsied 30 year old, Instant square may be the first opportunity to demonstrate spatial understanding and creativity.

MISTAKES I HAVE MADE WITH INSTANT

I have designed several Instant procedures and made numerous mistakes. For three and four year olds at the University of Virginia Child Development Center, I decided on a turtle control system so that the turtle moves were like points on a compass. Cursor control on Wordstar uses a logical key arrangement with E for up, X for down, D for right, and S for left. I adopted this spatial arrangement for an Instant program with four additional keys going northeast, southeast, southwest, and northwest.

Initially, I programmed it so that, with a single keystroke, the turtle turned and moved simultaneously. The resulting confusion over distance and direction probably interfered with the learning of turtle control. This confusion is apparent to me now; I think that separate keys for distance and direction should be used. For moves, 10 as a unit seemed too small for my audience; the square was barely discernible from the circle. I am still debating whether programming Instant squares, circles, and triangles is good for those children. I know they use them and make wonderful designs and interesting patterns, but wonder if they would do the same with only the turtle movements.

PROCESS AND PRODUCT

Other issues in Instant are whether a history of all the moves should be generated, whether procedures should be definable through Instant, and whether saving work should be built into Instant. If the objective for the children is control of the turtle, I see no reason for these being provided.

It must be recognized that this moves away from the process of Logo and toward a product. However, some students may need those single key abilities. Having an easily reproducible record can facilitate project orientation and problem solving. Procedures which save a history and allow the product to be redrawn or to undo the last move are probably very helpful. If the history was defined as a procedure, however, I would insist that the procedure be named meaningfully. Using numerals as procedure names (as in Delta Drawing) is counterproductive to Logo.

Five and six year olds often want to keep their work; Instant keys for READPICT and SAVEPICT might be written. To be consistent with Logo, the Instant program should ask for a name of the picture for labeling the picture on the disk. Even if only personal names were used, meaningful names are maintained.

INSTANT FOR UNDERSTANDING

Instant programs assume that the user will recognize the three part relationship of F, FORWARD, and forward turtle movement. At least two or three variations of Instant might be used which do not make this assumption about the learners.

Tom Lough's Slow Turtle procedures (NLX September 1983 and Teaching and Computers magazine, September 1983) could be built into the Instant program. When F is pushed, the turtle could move FORWARD 20 or 30 spaces, one space at a time. Another way to change Instant to emphasize what is going on is to have the computer print FORWARD 30 when F is pushed. These two additions are very simple variations to a standard Instant program.

```
TO START
   DRAW
   INSTANT
   END
```

```
TO INSTANT
   TEST RC?
   IFTRUE ACTION
   INSTANT
   END
TO ACTION
   MAKE "KEY RC
   IF :KEY = "F PRINT [FORWARD 20]
REPEAT 20 [FORWARD 1]
   IF :KEY = "B PRINT [BACK 20]
REPEAT 20 [BACK 1]
   IF :KEY = "R PRINT [RIGHT 15]
REPEAT 15 [RIGHT 1]
   IF :KEY = "L PRINT [LEFT 15]
REPEAT 15 [LEFT 1]
   END
```

The number of keys which are "alive" can be expanded to include any other needed commands. The size of the moves and turns can be changed. The point is to have a program which is explicit in what is happening when the key is pressed.

Another halfway place to consider is to have an Instant program which requires a return key to be pressed to execute the line. CHAR 13 is the ASCII code for RETURN. ACTION is changed so that after the command is printed, RETURN must be pressed to get the turtle to move.

```
TO ACTION
   MAKE "KEY RC
   IF :KEY = "F PRINT1 [FORWARD 20] F
   IF :KEY = "B PRINT1 [BACK 20] B
   IF :KEY = "R PRINT1 [RIGHT 15] R
   IF :KEY = "L PRINT1 [LEFT 15] L
   END
TO F
   IF RC = CHAR 13 REPEAT 20 [FD 1]
PRINT CHAR 13 STOP
   F
   END
TO B
   IF RC = CHAR 13 REPEAT 20 [BK 1]
PRINT CHAR 13 STOP
   B
   END
TO R
   IF RC = CHAR 13 REPEAT 15 [RT 1]
PRINT CHAR 13 STOP
   R
   END
TO L
   IF RC = CHAR 13 REPEAT 15 [LT 1]
PRINT CHAR 13 STOP
   L
   END
```

Another bridging idea comes from the British version of Instant Logo. The function keys on the ACORN microcomputer are programmed to print the commands FORWARD, BACK, RIGHT, and LEFT to which the input number is added from the keyboard directly. You gain the efficiency of typing without predetermining the size of the movements.

All of the versions are intended to make the connection between pressing a key and the result. None are substitutes for real Logo and many people will find them unnecessary diversions. You must make the judgement.

INSTANT ACCESS

Instant can remove the difficulty of using the keyboard for interacting with the computer and the turtle. Instant provides access to the computer to anyone who has difficulty with keyboard use. Two groups are immediately evident: young children and people with motor difficulty. Other special populations (autistic, EMR, and the like) might also use Instant's immediate interaction with the computer environment. The idea would be to provide a computer environment which is automatically responsive.

Alternate input devices, such as electronic pens, touch sensitive screens, and voice recognition systems also allow control of the computer. Instant can be considered to be a software appliance. For a teacher or worker with special populations, the possibilities of tailoring the keyboard to the particular learning needs are infinite.

However, Instant turtling is only one possibility in providing expanded access to computer functions. Instant drawing pads with a choice of designs (Koala Pad or Power Pad, for example), Instant message producers with a varied menu of messages, Instant calculators with demonstrations of the actual operation -- all kinds of Instant programs can be made for people who need the power of the computer to do interesting or important things for themselves.

Instant programs allow all of us access to the computer's power. These applications do not "teach." They merely are enabling environments. Word processing and accounting software might also be considered Instant programs which bring the power of those functions at the press of a key. Glen Bull and I have always contended that word processing and spreadsheet programs on the computer were Logo-like because they impart power to the user. Our definitions of Instant have been much too limited.

Tipps . .

INSTANT PROJECTS

All of these Instant ideas provide many projects for students and teachers. The keyboard can be transformed into any function. A key scramble (in which students change the letter printed in response to pressing a key) is a fun introduction to the Instant keyboard. Personal coders and encoders can also be designed. The attitude of Logo is, "I can make this computer do anything or become anything." Instant is an imperfect extension of Logo philosophy.

None of these projects are Logo. The same programs and appliances could be and have been written in other languages. The power of Logo is its adaptability to any use which can be dreamed up. Logo allows an everyday person (parent, student, teacher, business person) with a knowledge of Logo to create a responsive environment of any type.

(Ed. Note: For information on drawing pads which can be used with Logo, contact Koala Technologies Corporation, 3100 Patrick Henry Drive, Santa Clara, CA 95050, and Chalkboard, 3772 Pleasantdale Road, Atlanta, GA 30340.)

Steve Tipps is a professor in the University of Virginia's School of Education, and presents Logo workshops for school systems throughout the United States.

Chicago LLX News

Andy David has started a local Logo exchange for persons in the Chicago area. The first meeting was held October 15, and was attended by over twenty Logo teachers, teacher trainers, programmers, publishers, and consultants. Among the many ideas under consideration by the group are the establishment of a software exchange, test and evaluation site, and the regular publication of a many-to-many newsletter. Each newsletter subscriber sends in a short article of an agreed-upon length. The editor compiles all articles, prints them, and redistributes the copies to the contributors. For additional information, write Andy David, 5430 South Harper Avenue, Chicago IL 60615, or call (312) 684-3178.

MICROWORLDS

by
Glen Bull

Logo Grade Book

Part II Report Generator

In the last column, I discussed a method for using Logo to record and average grades. This month, I will review the use of list processing to generate a report of the recorded grades.

The form of the report ideally should resemble a page from a gradebook:

	#1	#2	#3	#4	AVG
John	89	94	ABS	91	91
Mary	88	ABS	79	86	84
Sam	87	91	93	88	90
Sally	94	95	92	100	95

ENTERING NAMES AND SCORES

In addition to the test scores for each student, a list of the names should be entered into the Logo gradebook:

```
MAKE "CLASS [JOHN MARY SAM SALLY]

MAKE "JOHN'S.SCORES [89 94 ABS 91]
MAME "MARY'S.SCORES [88 ABS 79 86]
MAKE "SAM'S.SCORES [87 91 93 88]
MAKE "SALLY'S.SCORES [94 95 92 100]
```

The class names and scores can be stored in a file as described in last month's column.

STARTING THE REPORT

The steps involved in printing the grades of a list of students involves the following:

- Print the name of the first student in the list
- Print the test scores of the student
- Print the student's average
- If there are any more students, go to the next one, otherwise stop.

The basic structure outlined can be tested with a short procedure to print all the names in the list of students.

Microworlds continued

```
TO PRINT.LOOP :CLASS
   IF :CLASS = [] STOP
   PRINT1 FIRST :CLASS
   PRINT.LOOP BUTFIRST :CLASS
   END
```

This procedure works after a fashion, although the result seems odd because there are no spaces between the names:

```
PRINT.LOOP :CLASS
JOHNMARYSAMSALLY
```

Nevertheless, the procedure does print all the names of the class, and then stop as we asked.

The next task is to ask the procedure to print the student's scores as well. In the immediate mode, the scores of the first student could be printed by typing:

```
PRINT :JOHN'S.SCORES
89 94 ABS 91
```

In order to make the procedure print the student's scores, it is necessary to ask the procedure to create a new word consisting of the student's name JOHN and the suffix 'S.SCORES, to form JOHN'S.SCORES. This task can be accomplished with list operator WORD, a built-in command in Logo. The command WORD is used to put letters together to form words. Before going on, it may be desirable to recall that:

- A <u>list</u> is a series of items enclosed between brackets. [JOHN MARY SAM SALLY]
- A <u>word</u> is an unbroken series of letters preceded by a quotation mark. "AVOCADO

Two words can be joined together to form a single word:

```
PRINT WORD "FIRE "TRUCK
FIRETRUCK
PRINT WORD "JOHN "S.SCORES
JOHNS.SCORES
```

This is close, but we still need the apostrophe.

PRINTING APOSTROPHES

The apostrophe is a reserved symbol in the MIT versions of Logo (Terrapin and Krell), and will not print directly because its use is reserved for other purposes.

The apostrophe is needed here to form the word JOHN'S.SCORES. Fortunately, each character has a number associated with it. Even when the character can not be printed directly, it can be printed by typing the Logo command CHAR in front of its number. For example, the number associated with the character A is 65, and the number associated with an apostrophe is 39.

```
PRINT CHAR 65
A
PRINT CHAR 66
B
PRINT CHAR 39
'
```

This approach can be used to create a word which will be used to access John's scores:

```
PRINT ( WORD "JOHN CHAR 39 "S.SCORES )
JOHN'S.SCORES
```

The parentheses are essential. The command WORD normally expects only two inputs. The parentheses override this expectation so that three or more inputs can be combined to form a single word.

There is one other point to note. In many cases, spacing around parentheses is not crucial in Logo. Here, however, there <u>must</u> be a space between "S.SCORES and the final parenthesis. Otherwise, Logo will think that the parenthesis is part of the final input.

THINGS AND NAMES

Now that the word "JOHN'S.SCORES can be formed, it is possible to access the THING associated with "JOHN'S.SCORES. THING is another Logo command which is the equivalent of the colon used to access the thing associated with a name. The following two commands are equivalent in Logo:

```
PRINT :JOHN'S.SCORES
89 94 ABS 91
PRINT THING "JOHN'S.SCORES
89 94 ABS 91
```

Since this is the case, THING can be used to access data associated with a name created through use of the list operation WORD.

```
PRINT THING ( WORD "JOHN CHAR 39 "S.SCORES )
89 94 ABS 91
```

This equivalent instruction is starting to have a cryptic appearance. It says to print the thing associated with word formed by combining "JOHN, and apostrophe (CHAR 39), and "S.SCORES. Another step is required before the instruction can be inserted into the procedure PRINT.LOOP. In PRINT.LOOP the name "JOHN is not accessed by typing the name directly. It is obtained by referring to the first item in a list of class names.

```
PRINT :CLASS
JOHN MARY SAM SALLY
PRINT FIRST :CLASS
JOHN
```

Therefore, typing FIRST :CLASS is the equivalent of typing "JOHN directly. This is shown below:

```
PRINT THING ( WORD "JOHN CHAR 39 "S.SCORES )
89 94 ABS 91
PRINT THING ( WORD FIRST :CLASS CHAR 39
"S.SCORES )
89 94 ABS 91
```

This new instruction, which is the equivalent of PRINT :JOHN'S.SCORES, can be inserted into PRINT.LOOP.

```
TO PRINT.LOOP :CLASS
   IF :CLASS = [] STOP
   PRINT1 FIRST :CLASS
   PRINT THING ( WORD FIRST :CLASS CHAR 39
"S.SCORES )
   PRINT.LOOP BUTFIRST :CLASS
   END
```

SPACING THINGS OUT

The revised procedure does print each student's name and test scores, but the formatting leaves something to be desired:

```
PRINT.LOOP :CLASS
JOHN89 94 ABS 91
MARY88 ABS 79 86
SAM87 91 93 88
SALLY94 92 95 100
```

One difficulty is that there is no space between each student's name and their first test score. CHAR 32 is a space, just as CHAR 39 is an apostrophe and CHAR 65 is an A. Therefore, printing CHAR 32 will produce a space in between the name and the test score. Although this instruction could be inserted directly into PRINT.LOOP, the creation of a procedure, SPACE, to print a variable number of spaces would be both more general and less cryptic.

```
TO PRINT.LOOP :CLASS
    IF :CLASS = [] STOP
    PRINT1 FIRST :CLASS
    SPACE 1
    PRINT THING ( WORD FIRST :CLASS CHAR 39
"S.SCORES )
    PRINT.LOOP BUTFIRST :CLASS
    END

TO SPACE :NUMBER
   REPEAT :NUMBER [PRINT1 CHAR 32]
   END
```

Inclusion of a space between the student's name and the test score helps the appearance of the format somewhat:

```
PRINT.LOOP :CLASS
JOHN 89 94 ABS 91
MARY 88 ABS 79 86
SAM 87 91 93 88
SALLY 94 92 95 100
```

LINING UP NAMES

It is now apparent that one of the reasons that the scores do not line up is the different lengths of student's names. One solution to this problem would be to require the students to acquire names of the same length. This solution was tried in the early days of computing, and met with some resistance!

Another solution would be to count the number of letters in each name, and insert enough spaces before each name to align the right edge of each word with the one above. For example, SAM needs two spaces before to ensure that the last letter in the name lines up with the final letter in SALLY. JOHN and MARY each need one space before them to line up with SALLY on the right-hand side. This type of alignment is called "right-hand justification," and is used in many publications, including the NLX.

A procedure called COUNT.LOOP can be created to count the number of letters in each name. COUNT.LOOP takes a word as an input, and uses BUTFIRST to remove letters from the word, one at a time. As each letter is removed from the word, a variable named :NUMBER is increased by one. When all the letters have been exhausted, COUNT.LOOP outputs the number of letters in the name. As a matter of convenience, a calling procedure LETTER.COUNT supplies a starting value of 0 for the input :NUMBER in COUNT.LOOP each time.

```
TO LETTER.COUNT :WORD
   COUNT.LOOP :WORD 0
   END
  TO COUNT.LOOP :WORD :NUMBER
    IF :WORD = "   OUTPUT :NUMBER
     OUTPUT COUNT.LOOP (BUTFIRST :WORD)
(:NUMBER + 1)
    END
```

The parentheses in the last line of COUNT.LOOP are only inserted for clarity, and can be removed without affecting the procedure. If the procedure LETTER.COUNT is asked to count the number of letters in "JOHN, it supplies an answer of 4.

```
PRINT LETTER.COUNT "JOHN
4
```

It is now possible to insert a line into PRINT.LOOP which will print six spaces, minus the number of letters in each name. For example, six spaces minus the four letters in "JOHN would result in the insertion of two spaces before the name. In this case, it is necessary to remember that FIRST :CLASS refers to the first name in the list of class names [JOHN MARY SAM SALLY].

```
 TO PRINT.LOOP :CLASS
    IF :CLASS = [] STOP
    SPACE 6 - LETTER.COUNT FIRST :CLASS
    PRINT1 FIRST :CLASS
    SPACE 1
    PRINT THING ( WORD FIRST :CLASS CHAR 39
"S.SCORES )
    PRINT.LOOP BUTFIRST :CLASS
    END
```

LINING UP SCORES

This refinement produces additional improvement in the appearance of the class scores when PRINT.LOOP is run. The same effect could have been produced with the cursor positioning commands in Logo. However, the cursor positioning commands only affect the screen, and do not format the text when it is sent to the printer. Therefore, the use of spaces to format the names will produce a more universal result which will work for both the screen and the printer.

```
PRINT.LOOP :CLASS
  JOHN 89 94 ABS 91
  MARY 88 ABS 79 86
   SAM 87 91 93 88
 SALLY 94 92 95 100
```

Now the names are all right justified, but the test scores are still not aligned. The reason for this is that the scores are not all the same width. Most are two characters wide, but a few are three characters wide. Although we easily could have used AB instead of ABS to indicate an absence, it would hardly be reasonable to ask Sally to accept a 99 instead of 100 on her last examination just to improve the formatting of the scores!

An alternate solution would be to insert spaces before some of the scores to line them up, just as we did with the students' names.

```
TO PRINT.SCORES :SCORES
   IF :SCORES = [] SPACE 1 STOP
   SPACE 5 - LETTER.COUNT FIRST :SCORES
   PRINT1 FIRST :SCORES
   PRINT.SCORES BUTFIRST :SCORES
   END
```

This procedure will insert three spaces before each of John's test scores, except for the third, for which John was absent. Only two spaces are inserted before the third test, to maintain an even alignment.

```
PRINT.SCORES :JOHN'S.SCORES
   89   94  ABS   91
```

The PRINT.SCORES procedure can now be inserted into PRINT.LOOP.

```
 TO PRINT.LOOP :CLASS
    IF :CLASS = [] STOP
    SPACE 6 - LETTER.COUNT FIRST :CLASS
    PRINT1 FIRST :CLASS
    PRINT.SCORES THING ( WORD FIRST :CLASS
CHAR 39 "S.SCORES  )
    PRINT []
    PRINT.LOOP BUTFIRST :CLASS
    END
```

This revision takes care of the remaining formatting details in PRINT.LOOP. The names and scores are now aligned when the procedure PRINT.LOOP is run.

```
PRINT.LOOP :CLASS
  JOHN   89   94  ABS   91
  MARY   88  ABS   79   86
   SAM   87   91   93   88
 SALLY   94   92   95  100
```

AVERAGING THE SCORES

The remaining task is to print the averages as well as the scores. This can be accomplished with the procedure to average lists of numbers developed in last month's column.

```
PRINT AVERAGE :JOHN'S.SCORES
91.333
```

In the formatted grade listing, an average rounded to the nearest integer is desirable. This can be accomplished with the Logo primitive ROUND.

```
PRINT ROUND AVERAGE :JOHN'S.SCORES
91
```

This method of printing the average can be inserted directly into PRINT.LOOP.

```
 TO PRINT.LOOP :CLASS
    IF :CLASS = [] STOP
    SPACE 6 - LETTER.COUNT FIRST :CLASS
    PRINT1 FIRST :CLASS
    PRINT.SCORES THING ( WORD FIRST :CLASS
CHAR 39 "S.SCORES  )
    PRINT ROUND AVERAGE THING ( WORD FIRST
:CLASS CHAR 39 "S.SCORES  )
    PRINT.LOOP BUTFIRST :CLASS
    END
```

Do not feel that this is the ultimate version. In fact, there are many improvements which can be made. However, I hope that you have gained a fuller understanding and appreciation of the power and capabilities of the word, list, and arithmetic sides of Logo in the process.

LCSI VERSION

Some of these procedures for LCSI Logo do not seem to work unless they are defined in the editor using EDIT, rather than TO. I am at a loss to explain this, but, until the reason is determined, I suggest that you use EDIT.

Note that, in order for PRINT.LOOP to work, the procedures AVERAGE.LOOP and AVERAGE from last month's column must be included.

```
TO PRINT.LOOP :CLASS
   IF EMPTYP :CLASS [STOP]
   SPACE 6 - LETTER.COUNT FIRST :CLASS
   TYPE FIRST :CLASS
   PRINT.SCORES THING WORD FIRST :CLASS
"'S.SCORES
   SPACE 3
   PRINT ROUND AVERAGE THING WORD FIRST
:CLASS "'S.SCORES
   PRINT.LOOP BUTFIRST :CLASS
   END
TO PRINT.SCORES :SCORES
   IF EMPTYP :SCORES [SPACE 1 STOP]
   SPACE 5 - LETTER.COUNT FIRST :SCORES
   TYPE FIRST :SCORES
   PRINT.SCORES BUTFIRST :SCORES
   END
TO SPACE :NUMBER
   REPEAT :NUMBER [TYPE CHAR 32]
   END
TO LETTER.COUNT :WORD
   OUTPUT COUNT.LOOP WORD 0
   END
TO COUNT.LOOP :WORD :NUMBER
   IF EMPTYP :WORD [OUTPUT :NUMBER]
   OUTPUT COUNT.LOOP BUTFIRST :WORD
:NUMBER + 1
   END
```

Glen Bull is a professor in the University of Virginia's School of Education, and teaches Logo courses at both the graduate and undergraduate level.

NOVA's "Talking Turtle" — A Compelling Documentary

Reviewed by Andrew David

"Talking Turtle" was the subject of the weekly science program NOVA, broadcast on October 25th on public television stations nationwide. For many, I'm sure, this sensitive documentary furthered the exciting exploration into Logo's larger educational context.

Many of us began this exploration by reading Seymour Papert's book, <u>Mindstorms: Children, Computers, and Powerful Ideas</u> (BASIC Books, 1980). Here, in his eloquent treatment of the pressing subject of computers and education, Papert wrote of an educationally rich culture in which individuals may take control over their own learning. His vision was both hopeful and challenging.

Since then, the task befalling parents, teachers, and children has been a less attractive one of implementation. When do we use the quotes and when do we use the colon? Which version will best serve our purpose? What about lists? These considerations have, in part, distracted us from the forceful social concerns that have motivated Papert and others of the MIT team that developed Logo.

NOVA continued

Writer Suzanne Campbell-Jones and a competent production team have captured the power of Logo through a series of profiles highlighting its diverse applications. Among those subjects examined were the original floor turtle robot (from which the familiar screen turtle has descended); disabled students who, through Logo, are experiencing a rare opportunity to explore and manipulate space; the Lamplighter Project of Dallas, Texas, boasting America's highest computer to student ratio; an inner city project, where otherwise "non-exceptional" students excel on the computer; and several bilingual versions, including one in Wolof, the native language of Senegal.

These scenes of very real application have been interwoven with pertinent interviews, and, of course, the ideas and the urgings of Seymour Papert. He is as articulate a speaker as he is a writer; his passionate manner and constant use of non-computer metaphors remind us that his focus on computing machines has done little to dilute his original emphasis on people and their needs.

"Education," he tells us, "is much more related to love than to logic. Education has very little to do with explanation - it has to do with engagement." Every moment of the film supports this. A child who can barely control the movement of his hands takes three carefully calculated actions and causes the floor turtle to move a tower of blocks; a young but sophisticated programmer watches her program "fail," then immediately tracks down the problem and corrects it; native Senegalese children program the turtle to draw pictures of huts ... <u>their</u> huts, images out of their own lives.

This is welcome testimony to the variety of experience that is reflected in Logo programming! With equal sensitivity, the filmmakers have shown that Logo's problem solving approach is part of the living, breathing world outside the computer screen as well. Scenes of "turtle play" in a pile of leaves, young dancers spinning on the floor to choreography that has been scribed in a "procedural" format, gears and jugglers (two of Papert's most important metaphors) have all been delicately chosen to enhance the assertions that Logo is not about binary addition and cathode ray tubes, but about living and learning.

"Talking Turtle" was presented in the true spirit of Seymour Papert and his vision. Of course, it is possible to use Logo in a very different way than that which he envisions. But to deprive Logo education of the insights and social concerns from which it was spawned - insights illuminated by such works as <u>Mindstorms</u> and now this NOVA feature - is to insure that its potential will never be realized. As teachers and programmers, we are all struggling to determine <u>how</u> to use Logo. This outstanding production reminds us <u>why</u>.

Andrew David is doing Logo networking and advanced Logo applications research in Chicago. He also works with elementary students citywide in weekly workshops on Art and Perception.

NLXual Challenges

with Robs Muir

Starring Logo

If you have spent some time reading Logo manuals and instructional materials, you have played around with POLY. POLY is very good for drawing regular polygons. All regular polygons have one important characteristic; when the turtle finally returns to its starting position, it has turned 360 degrees (no more and no less). One of the interesting aspects of POLY is that you can generate not only regular closed polygons (e.g., square, triangle, hexagon, etc.), but you can also create complex polygons.

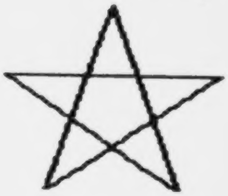
A complex polygon is a closed figure (the turtle returns to its starting position and heading) with a total turning of some integer multiple of 360 degrees. Most of us are familiar with a Logo "standard," the five pointed star. However, we usually introduce our students to the complex polygonal form of a star created by

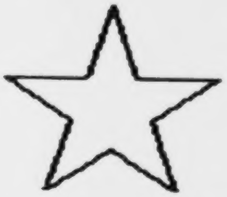
```
REPEAT 5 [FD 50 RT 144]
```

This clearly illustrates a total turning of a multiple of 360 degrees (5 * 144 = 720 = 360 * 2).

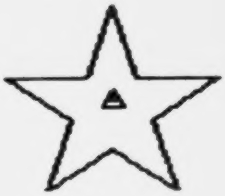
This month's problem is built around such a five pointed simple polygon. Our first challenge is to build a star in which no one side crosses another; the total turning is exactly 360 degrees.

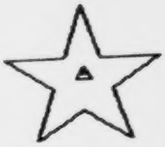
Challenges continued





Just to make things a bit more difficult, have the turtle start and finish in the exact center of the star. Then, write a procedure which will draw the star of any determined size with a single input.





Once this is accomplished, you are ready for the remainder of the challenge. Draw a five pointed star, and then "stamp" a second star, using the same axis of rotational symmetry. Try different sizes, and different angles of rotation also. Can you make a space filling pattern out of the star and other geometric shapes?

The ancient Greek mathematicians believed that the simple star embodied the deepest knowledge of geometry. Contained within the star is the "golden mean," the basis of much Greek art and architecture, and the pentagon, the mystic symbol of the Middle Ages. Our double star image is also a commonly recurring theme in Islamic art; its complexity and beauty is revered in Middle Eastern decorative art.

The underlying difficulty of this problem may not be readily apparent when you first begin. If you have really plumbed the depths, you should be able to pack the double stars anywhere on the screen in any size. If you have some initial difficulty, remember that humans have studied this figure for at least 25 centuries.

As with all NLXual Challenges, if you want to receive a possible solution, send a copy of your solution (or your attempt) with a letter size self addressed stamped envelope to NLXual Challenges, Attn: Robs Muir, 1688 Denver Avenue, Claremont CA 91711. We will compile contributions and send you the most elegant solution.

If you would like to submit a challenge or puzzle for this column, send your ideas to the above address.

Robs Muir is a teacher in the Claremont CA Unified School District and Claremont Graduate School. He presents Logo workshops in the southern California area.

Logo Publication Announced

TurtleTalk is a quarterly newsletter published by Larry Lohrman, of Issaquah, WA. Featured are articles about Logo teaching and learning environments, reviews of Logo versions, explanations of procedures, and examinations of Logo related subjects, such as artificial intelligence and robotics. Larry offers TurtleTalk subscribers a 10% discount on popular Logo books. Subscriptions to TurtleTalk cost $15 per year (4 issues) for the USA, Canada, and Mexico, $20 per year elsewhere. For a free sample copy, write: TurtleTalk, 955 Greenwood Boulevard, Issaquah, WA 98027.

The National Logo Exchange, copyright 1983 by Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly, September through May, $25 (US) per year, mailed FIRST CLASS from Charlottesville, Va. $5 (US) additional per year for addresses outside the USA, Canada, and Mexico. The opinions expressed by the authors are not necessarily those of The National Logo Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress Street, Salem, MA 01970. Address other correspondence to: The National Logo Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor....................................Tom Lough

ISSN 0734-1717

The National LOGO Exchange

Volume 2 Number 5 FORWARD 100! January 1984

The Hazards of Hacking

by

Sharon Burrowes

Logo "slipped" into our school system last year via our Talented and Gifted (TAG) program. As our resource teacher introduced it to groups of 5th and 6th grade children at each school, the interest in the computer (and especially in Logo) spread like wildfire. Before we knew it, instead of having one computer for the TAG program that rotated from school to school, we found ourselves with one computer in each school!

Of course, this created questions such as, "What is the curriculum?" and "How do teachers get training?" which the school system set about dealing with. Meanwhile, the eager Logo loving students were pounding away at their keyboards, creating fascinating works of "turtle art."

This fall, as part of our newly developed system-wide curriculum, we decided to sponsor a series of "training sessions" for selected 6th graders who were last year's most enthusiastic Logo-ers. The sessions were to focus on the non-graphics aspects of Logo so that these students could grow in a different direction. By the second of the four sessions, it became apparent that two bad habits had developed in most of these students during their months of "hacking" last year.

HACKER HABITS

The first of these was the tendency to take a model procedure, key it in, run it, and dash onward to the next model. Little attempt was made to understand why the procedure functioned as it did. They seemed driven to try something new, ooooh and ahhh at the results, and then plunge on the next example - thus merrily skimming the surface.

While I constantly encourage teachers to let students "mess about" and explore, it is apparent that students need guidance to explore in more depth. Without depth, they seem to get forever stuck on FORWARD, BACK, RIGHT, LEFT, and a few other turtle control commands.

For example, in the first session, I had used the statement MAKE "NAME READLIST at one point in a procedure. Later in the same procedure, I used PRINT :NAME. In the next session, I introduced an example which contained the statement MAKE "STUFF WORD "HI "THERE. Most of the students said that this MAKE statement would cause the computer to print out HITHERE. This seemed to indicate that they had never really experimented with the statements from the previous week, in spite of having computer time during which they were supposed to do exactly that!

In another instance, I had suggested to a group that they simulate the roll of a die using RANDOM. The next week, they presented me with a procedure which I knew was beyond their capabilities. They admitted they had copied it and did not understand how it worked. They had not experimented with the idea sufficiently.

BREAK IT UP

My second concern will be no surprise to anyone who has watched students write their first turtle graphics procedures. The procedures get longer and lonnnger and lonnnnnnger! I found that students had not been shown and did not spontaneously invent the concept of modularity.

A pair of students had written an exploratory procedure using such things as REPEAT 1000 [ARCRIGHT 50]. The procedure produced fascinating designs on the screen, ending with what looked like stars winking out. The students wanted to have the turtle write END when this was all done, but their E had an extra line in it. (These students has not yet been shown the NLX ABC's.)

Hazards continued

The procedure took a minute or so to run up to that point; thus, they were not getting very far with isolating the bug. I showed them how to break their existing procedure up into modules and write a superprocedure procedure to call the other parts. They were delighted. One student popped out with, "Is that why our procedures won't work right?" This question, it turned out, really meant, "Is that why we can't GET our procedures to work right?"

In another group were two boys who had written a long procedure which our TAG coordinator sent to the NLX. Editor Tom Lough broke the procedure into subprocedures, and sent it back for them to study.

When I began talking about keeping procedures small, one of these boys blurted out that their procedure had been sent to "some man" and he sent it back broken up into smaller parts! They seemed very impressed that "some man" and I came up with the same idea! In fact, it seemed to be enough to convince them that there really was something of value to the idea.

The concept of subprocedures needs to be introduced fairly early. Long procedures are frustrating to debug. But, by taking "masterpieces" which have bugs in them and cleaning them up with the students, the ease of debugging is clearly demonstrated.

In hindsight, we realize that we should have emphasized exploration of commands and their multiple uses more and modularity more than we did both in working with students and in training teachers. However, if you find yourself in a situation such as we have where Logo lovers have rushed ahead of the teacher training and curriculum, do not despair. It does seem possible to go BACK 50 and LEFT 45 to get students and teachers pointed in the appropriate directions. Better programming habits will spread among the children when they help us be more effective.

Sharon Burrowes is a consultant and a computer science teacher in the Wooster (Ohio) City School System, and a frequent contributor to the NLX.

From The Editor

In November, 1982, the staff of The Computing Teacher devoted an entire issue of their magazine to Logo. It was an outstanding effort, and made a significant contribution to helping Logo teachers everywhere.

They have replicated their effort a year later with the recent December / January issue. It contains many useful Logo articles, including 4 separate treatments of recursion!

David Moursund's editorial, "Logo Frightens Me," deserves special mention. Moursund is a respected and experienced computer educator. He is a strong supporter of Logo, but harbors some fears which are worth thinking about.

Is Logo being oversold? Are overzealous educators reading Mindstorms, buying Logo, and developing unreasonable and unattainable expectations of what Logo can do in their classrooms? To view Logo as the "ultimate answer" in computing education is folly.

Will Logo reach its full potential in our classrooms? How will we know if it makes a difference? Research reports now available do not provide sufficient information on Logo's effectiveness with "ordinary" teachers - those with a modest level of training, experience, and interest in the computer field.

This brings Moursund to the heart of his fear - that the importance of full teacher support will not be recognized. Providing training, experience, resources, and research opportunities for the teachers as well as continuing computer access for students must be high on priority lists if the full potential of the Logo language is to be realized.

Moursund's cautions deserve discussion and thought. We agree with his suggestions and recommend that his editorial be carefully considered.

Another key article, "Problem Spaces in a Project Oriented Logo Environment," by the NLX's Steve Tipps and Glen Bull, has been copublished by the NLX, with Part II appearing in this issue.

Editor continued

Jim McCauley's outstanding article, "Kepler," should be required reading for all Logo teachers. McCauley describes his interactions with a group of children as they struggle to use Logo to help them study planetary motion. His sensitive description of the ups and downs of their discovery process will help all readers to identify more effective ways to facilitate Logo learning. His candid admissions of his own self doubts serve as mirrors for us all, and offer encouragement through his personal example.

For information on obtaining a copy of <u>The Computing Teacher</u>'s Logo issue, write to Sharon Franklin, TCT, University of Oregon, 1787 Agate Street, Eugene OR 97403.

For Logo, the last twelve months have seen a groundswell of interest and development. The depth and breadth of ideas in TCT's Logo issue demonstrates how far thinking about Logo has come in 1983. And we're just getting started! We at the NLX wish for you and your students much growth and excitement with Logo in 1984! Happy New Year!

```
MAKE "84 100
FD :84
```

Logo Down Under

Australia has quite a large active Logo population. We recommend that NLX readers consider subscribing to <u>COM 3</u>, an Australian magazine published 5 times per year by the Computer Education Group of Victoria (CEGV). Here is a partial listing of titles from a recent issue. "Using Multiple Turtles to Create Games," "Turtling On: Excerpts from some Turtle Tales," "Turtle Talk," a regular column by Sandra Willis, and a review of Prentice Hall's <u>Let's Talk Apple Turtle</u>. For more information, write <u>COM 3</u>, 32 Chrystobel Crescent, Hawthorn, Victoria, Australia, 3122.

MECC Announces New Logo Products

Two Logo products have been released by the Minnesota Educational Computing Consortium (MECC) to help teachers with Logo in their classrooms.

"Introduction to Logo for Teachers" contains training materials developed from the combined efforts of Logo teacher trainers throughout Minnesota. Its 116 pages contain instructor notes, transparency masters, activity pages, and reference materials. The book and accompanying diskette were designed to be used with Apple (LCSI) Logo. The instructor notes consist of nine learning module segments, with objectives. Teachers using the material are cautioned to avoid over-structuring the use of Logo. The price is $49.00.

"EZ Logo" is a set of Apple (LCSI) Logo learning materials designed to help children of primary age (4-8 years old) learn to use the computer in self directed ways. Based on a version of INSTANT, EZ Logo was developed by Judith Anderson, and was an award winner in MECC's 1982 Teacher Programming Contest. Two levels are presented. Included in Level 1 are 24 turtle activities in module form, with supporting materials, command cards, activity guides, and a set of stickers for the computer keyboard.

Level 2 includes activities to teach how to begin combining Logo skills for more complex accomplishments, and to make copies of pictures. The materials are not intended as a substitute or replacement for Logo. Rather, they provide very young children with "EZ" ways to explore many aspects of turtle graphics. The 79-page manual and accompanying diskette sell for $48.00.

To order, send your check or purchase order to MECC, 3490 Lexington Avenue North, St. Paul MN 55112. For more information, call (612) 481-3500. Add 10% for orders from addresses outside the continental United States.

Problem Spaces in a Project-Oriented Logo Environment

by Glen Bull and Steve Tipps

Ed. note: This article is also appears in the December / January issue (Logo issue) of The Computing Teacher, and is published here in two parts. In part I, appearing in last month's NLX, a two dimensional matrix was set up to define four possible problem spaces in a Logo environment.

Implementation (Solution)	Source of the Project	
	LEARNER	FACILITATOR
LEARNER	I	III
FACILITATOR	II	IV

Learner Generated / Learner Implemented (Type I) and Learner Generated / Teacher Facilitated (Type II) Problem Spaces were explored. In this continuation, the remaining problem spaces will be explored.

Part II

III. Teacher Generated / Learned Implemented Problem Spaces

Some children invent their own projects, while others languish for lack of direction. One reason for tiring of Logo is a low skill level. Another is lack of interesting examples. Potential solutions for ennui in Logoland are ideas which build skills or provide models for exploration. The teacher generates a project for student solution in a Type II problem space.

A Type III project can become a static model. Asking students to type in a POLYSPI procedure which they don't understand is not a valid Type III project. The problem is not with POLYSPI, but with the short circuiting of the learning in POLYSPI through imposition of a solution. Teachers grounded in the purpose of Logo do not require a rigid scope and sequence chart with fixed activities; besides, most students already receive enough practice in doing things they don't understand.

Teachers do need many open ended ideas and projects. Characteristic of a good Type III project is a clear objective which can be reached in a variety of ways. A didactic approach can involve a specific suggestion:

"Place a square inside a circle."

A less structured approach could include the suggestion for a range of projects appropriate for the learner, using one procedure as an example:

"This procedure places a diamond in a circle. Can you create other shapes which fit inside? What other shapes could fit inside each other?"

Long range cooperative projects may be even less structured. The term "problem schemes" has been invented to describe long range classroom projects which emphasize applications of Logo in content areas. A class project to create a variety of increasingly sophisticated programs to generate random sentences according to linguistic rules is an example of a problem scheme.

In this instance, the teacher has deliberately chosen a context which requires deliberation on the nature of English syntax. The incongruities generated will inevitably create questions about the ways in which words work together. Such projects have an ideal Logo flavor, because the ideas sparked can lead in a variety of directions. This situation may be as threatening as a non-authoritative role for some teachers. The teacher directing the project must have a firm grasp of the potential subject matter embedded in the project.

Teachers need explanations of how content embedded in a project can be developed in a classroom setting. Discussions of subject matter are missing; the emphasis is on programming concepts required to conduct a project. For example, ANIMAL is a well known program in which the computer attempts to guess an animal by asking questions such as:

*Does the animal have four legs?
*Does it bark?

The program makes a final guess about the animal, asking: "Is it a frog? Or, is it an elephant?"

Problem Spaces continued

This is a potential problem scheme for the classroom. When developing this project, a class will discover why some nouns have the article "a" before them, while others have the article "an." Members of the class can become linguists, attempting to discover patterns which govern sentence structure. Questions arise: "Why do vowels exist?" "What is a vowel?" In this type of project, the class confronts natural rules found in the English language, rather than disassociated rules memorized by rote from a workbook.

The greatest potential of Logo in a classroom is its possible use for elimination of disassociated learning. This potential will be neglected as long as extensive illustrations of how Logo can be used to learn specific content are not available.

IV. Teacher Generated / Teacher Implemented Problem Spaces

The teacher supplies both the idea and the implementation of the project in a Type IV situation. At first glance, Type IV problems seem to be at odds with Logo philosophy, but problems of this sort may be worthwhile to illustrate a new capability of the language or to seed new student generated problems.

If a student has seen only the turtle graphics, a program which uses list processing to manipulate English or to graph a linear equation may be a revelation. Music generators can provide "hooks" for students to create procedures for their own projects. Tom Lough's NLX alphabet procedures use the turtle to draw letters which can be used in student projects.

Templates provided by teachers also may be a Type IV project. A poetry formatter which asks for words to complete a haiku or cinquain form might entice students to explore those forms. The questions, asked in a fill-in-the-blanks way, make apparent the form of the poetry.

When teachers first learn Logo, they often gravitate toward another kind of Type IV project - the drill-and-practice or tutorial programs, which are easy to write in Logo. Teachers develop their own skills in programming Logo with such projects, but using them in class to teach content is inherently anti-Logo and has nothing to do with Logo for student use.

A Call for Type III Projects

Valid projects from all four problem spaces should be used in a complete Logo environment. Logo's potential is reduced if it is restricted to one type of problem. Type I problems permit free exploration and play. Type II problems invite teacher-learner collaboration. Type IV projects expand the use of Logo in areas which exceed the learner's programing skills. Type III projects are valuable because they are particularly suited to the use of Logo in learning subject matter.

Logo with Content

Where are the projects in Logo which focus on learning experiences in algebra, social science, industrial arts and other content areas? The Logo community has avoided describing Logo projects of this kind for fear that descriptions might become recipes - explanations that shift into rigid lockstep programs deemed to be "the right way to do Logo" or "the right thing to do with Logo." A written description of a project tends to capture product rather than the process. It is a still picture rather than a movie.

Some teachers do not recognize that using Logo to teach a subject is even a possibility. A high school math teacher asked, "Can Logo help me teach algebra?" Other teachers realize the potential but find it difficult to develop methods of using Logo to teach specific concepts. How can Logo be used to understand multiplication of polynomials? Traditional methods have focused on mechanical techniques which can be used to obtain the answer without understanding the process.

For example, the solution to the following operation can be determined with the mnemonic FOIL (First, Outer, Inner, Last) which specifies all possible paired combinations of the elements:

$$(5X + 3)(2X+1) \text{ becomes } 10X^2 + 11X + 3$$

The teacher who attempts to present this concept through Logo is first tempted to create a tutorial with Logo. After recognizing the discrepancy between the intent of Logo and this tutorial approach, the second effort may involve the use of Logo to create a graphing tool. This then becomes a Type I project for the teacher, but is a Type IV

Problem Spaces continued

tool for students. From the student's perspective, it doesn't really matter whether the graphing tool is written in assembly language, BASIC or Logo. Calling the graphing tool a microworld doesn't alter the situation.

A Proposed Model

What is needed is a project which will allow students to use Logo directly to explore the meaning of polynomial multiplication. One approach involves a geometric method utilizing turtle graphics. A second technique might be based on exploration of the distributive law using list operators. Several Stages are likely before a Logo project actually reaches the classroom. Implementing content - laden projects may include:

* Recognition of the potential of using Logo for learning specific concepts
* Identification of a concept
* Determination of avenues of attack with Logo
* Development of the problem scheme
* Use in the classroom

The fourth stage involves the design of an experience in teaching without telling. **A true Logo problem demands both inquiry and discovery.**

A model problem scheme in the Type III category (teacher generated / student implemented) should contain a complete English description of the problem with a variety of possible directions in which the project might go. Exploration of area is one possible direction for binomial multiplication, but others might be suggested. The teacher needs an understanding of the concepts potentially embedded in the project. Given the nature of Logo, this frequently includes more than one subject area. With the distributive approach to binomial multiplication, compound words and sentences as well as combinatorial theory are outcomes.

The project would not include a full programming implementation to reduce the temptation to use the project in a static, cookbook - like fashion. At most, key programming ideas or program stubs might be suggested. The challenge to the Logo community is to provide projects which involve learning about content or subject matter without resorting to direct instruction with the computer as a teaching machine. Logo should make the computer a learning machine.

Problem schemes in many subject areas are essential if Logo is to go beyond computer literacy. This is a complex task - we still do not have fully realized Logo projects for binomial multiplication and have only arrived at our present understanding after several weeks of thought and sharing with high school mathematics and science teachers with strong Logo backgrounds. No single teacher will have the resources to create a series of projects spanning a school year. The challenge to the Logo community is to jointly develop and share ideas for projects of this kind.

Steve Tipps and Glen Bull are professors in the School of Education at the University of Virginia, and are regular columnists for The National Logo Exchange.

Be a Pal

Many Logo classes throughout the United States and Canada are participating in the NLX's Logo Class Penpal Network. Classes are matched by grade level, Logo version, and experience.

The following Logo classes still need to be matched with counterparts.

* Cyberlogo class of 30 4th and 5th grade students.

* Apple Logo and Commodore Logo class of 20 kindergarden children.

* Radio Shack TRS-80 Color Logo class of 33 4th graders.

If your situation corresponds closely to one of these, won't you consider participating? Send a stamped self addressed envelope today to NLX, Attn: Logo Penpals, Box 5341, Charlottesville, Va 22905.

Can You Help?

Help wanted! Peggy Goold uses Terrapin Logo on her Apple II+. She cannot get her Star Gemini-10 printer and APIC/G interface to function with it. Any NLX readers who have experience with these peripherals or other related ones please write to Peggy with suggestions. Peggy Goold, PO Box 264, Averill Park, NY 12018.

TIPPS for TEACHERS

by
Steve Tipps

Triangle thinking: Side 1

Anyone who has worked with Logo for more than three weeks has probably discovered that turns of 120 rather than 60 are needed to make an equilateral triangle. Experience with other triangles provides many new insights about geometry and many new skills in Logo.

My columns for January through March will be about "triangular thinking," and will suggest many different projects using triangles. Activities can be approached at levels from turtling with procedures through axioms and postulates in geometry and trigonometry. The learning intent and outcome of these activities will vary from class to class.

PYRAMIDING

Equilateral triangles in pyramid grids and patterns are a good exercise in variable procedures and positioning logic.

```
TO TRIANGLE :SIZE
   REPEAT 3 [ FORWARD :SIZE RT 120 ]
   END
```

Triangles can be organized with moves and turns coordinated for different arrangements. A simple pyramid might be two triangles at the base with another on top. The network could grow with three on the first level, two on the next level, and one on top; then four, three, two, and one on top. The repeated structure of the pyramid may incite the old question, "Isn't there a better, easier, simpler way to do this?"

AN EASIER WAY

Building a module called TRIANGLE.ROW is one easier way to do this. Two variables would be needed: the :SIZE of the triangle and the :NUMBER of them. A repeat statement could be used to draw the triangle, move over, draw another, etc. Positioning is crucial. The turns and moves have to be coordinated with TRIANGLE to produce a row.

To get the triangle base horizontal, a RIGHT 30 is put in to TRIANGLE. The turn works well for one triangle, but leads to complications with multiple triangles. "State transparency" helps simplify the case. This means that the turtle returns to its beginning position and heading after completing each drawing. That way, you know exactly where it is and which direction it is facing at the completion of each triangle. This makes moves much easier to plan. Here, state transparency is maintained by putting in a compensating LEFT 30.

```
TO TRIANGLE :SIZE
     RT 30
     REPEAT 3 [ FORWARD :SIZE RT 120 ]
     LT 30
     END
```

Positioning can be handled outside the procedure in the immediate mode or separate positioning moves in a master procedure.

MODULARITY AND POSITIONING

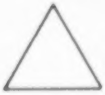
How positioning moves are handled seems to indicate the level of understanding of modularity. The less sophisticated problem solver will put all positioning moves into the procedure itself. This limits the usefulness and flexibility of procedure. The second stage is to keep the positioning separate from the procedure so that a shape can be moved or oriented according to the specific need of the design. The last step in the process is to make positioning procedures which can be used flexibly to move the turtle almost automatically.

```
TO MOVE.OVER :SIZE
     PU RT 90
     FD :SIZE
     LT 90 PD
     END
```

MOVE.OVER can be used to move right or left. Positive numbers scoot the turtle right while negative numbers scoot left. Another procedure gives flexibility of angle as well.

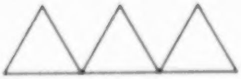
```
TO MOVE :SIZE :ANGLE
     PU RT :ANGLE
     FD :SIZE
     LT :ANGLE PD
     END
```

Tipps continued

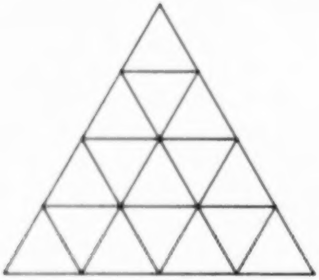


With this arsenal of procedures, a pyramid is easily constructed. One possible English description would be:

```
To make a row of triangles
     Draw a triangle
     Move over the length of the base
     Draw another triangle
     Move over, draw another
     Continue until the row is finished
     Move back to the first position of
        the turtle
```



```
To make a pyramid
     Draw a row of triangles
     Move up for the next row
     Draw a row with one less triangle
     Move up for the next row
     Draw a row with one less
     Continue until a row with only one
       triangle is drawn
```



In these descriptions, positioning moves are used as much as actual drawing moves. The eventual pyramid procedure can be written calling only simple procedures and REPEAT; it can also be written with recursion and testing. The level of implementation depends on the skill of the programmer, but the thinking, planning, and describing is challenging for all.

PYRAMIDING ACTIVITIES

The teacher's role is to create an environment which invites the creation of pyramids of many different kinds. Pyramids can be made with different size or color triangles on each row. The pyramid can be drawn from the top or the bottom. The pyramid can be rotated around the screen in a SPIN.PYRAMID procedure. Pyramids can be stacked together to make a pyramid of more pyramids.

Fortuitous mistakes may create other uses for TRIANGLE and PYRAMID. The screen may fill with equilateral triangles of different sizes and shapes. Rows of triangles of decreasing, increasing, or random sizes will be invented. Various combinations of colors, shapes and sizes may become pattern challenges. Designing wall paper or wrapping paper with triangles is a good activity.

Another patterning activity has a child creating a sequence (for example, purple square, blue circle, orange triangle, blue square, orange circle) for others to finish. Sizes can be added for a pattern with three elements. One activity with pyramids is to find how many different ways the pyramid lines can be traced without going over any line twice. The solutions can be worked off computer, then programmed quickly to check. The challenge from the teacher could be to discover how many possible paths there were.

AUTOMATIC TRIANGLE

Extend triangular understanding with an exploratory activity on right triangles called Automatic Triangle. Automatic Triangle takes advantage of the line drawn when the Turtle goes HOME. Type a distance move, a RT 90 or LT 90, another distance move, and HOME.

```
FORWARD 100
RIGHT 90
FORWARD 40
HOME
```

```
FORWARD 67
LEFT 90
FORWARD 42
HOME
```

more Tipps

The sequence of four commands will always result in a triangle. Students can speed the process with an AUTO.TRIANGLE procedure and two variables for :SIDE1 and :SIDE2. The exploration can be extended by using RANDOM to generate the lengths of the two sides.

MEASURING SIDE THREE

The length of the third side can be measured on the screen with a flexible clear ruler and recorded on a chart.

	Side One	Side Two	Side Three
#1	100	40	8.3 cm.
#2	60	60	6.5 cm.

The issue of scale will come up. Should you change turtle steps to centimeters or centimeters to turtle steps? Different monitors may have slightly different conversions factors to consider. We use the ratio of 13 turtle steps to 1 centimeter on our own monitors.

Students will notice that side three is longer than side one or side two. The teacher can challenge them to to find an arithmetic rule for the length of side three. Simple addition gives a result which is too large. Students may also try adding a fraction of one side to a fraction of the other side. This will work for some combinations, but not for others. Although this discovery method is sometimes slow and frustrating, the students are on the right trail, the trail of Pythagoras.

While following the trail, a number of very interesting side trips are possible using AUTO.TRIANGLE. This procedure can be used to explore the ideas of vector composition in physics and slope (gradient) in mathematics. What happens if you add 10 to one side and subtract 10 from the other side, for example?

```
AUTO.TRIANGLE 90 10
AUTO.TRIANGLE 80 20
AUTO.TRIANGLE 70 30
AUTO.TRIANGLE 60 40
```

The same process can be done with the lengths of the sides incremented recursively. A triangle depth projection effect is the result. A question for further exploration might be how to make a circular design with AUTO.TRIANGLE.

PYTHAGORAS' TRIANGLE TRAIL

Pythagoras and his theorem about the length of the third side can be introduced easily with Logo. Working out a teaching activity with the Pythagorean theorem and Logo was a project which NLX subscriber Maro Medovic-Zandee of Santiago, Chile, did in our Logo class last summer at UVA. Maro's procedures were especially beautiful because they were in Spanish!

As students work on the chart of sizes, some may try multiplying :SIDE1 by :SIDE2. A hint about multiplying a side by itself may be needed. Logo might be used directly for constructing a square on each side of several triangles. AUTO.TRIANGLE lets you make triangles of any size, but easy triangles such as those with sides of 60, 80, and 100 are useful for simplifying calculations. The general formula for square area is side times side. The chart can be expanded to hold this information as well, after the centimeters are converted to turtle steps, for example.

	Side One	Side Two	Side Three
#1	100	40	108
Squared	10000	1600	11664
#2	60	80	100
Squared	3600	6400	10000

When students start looking for a relationship, they usually will find one. In this case, "the square (area) on the long side is equal to the sum of the squares (areas) on the other sides." Short hand for this is

$$a^2 + b^2 = c^2$$

The terminology "squared" has a real meaning for the process of multiplying a number by itself. The geometric property of area is related to arithmetic squaring. The Logo primitive for square root is demonstrated as being the answer to the question, "What number multiplied by itself is 100? (or 245, or 1600394)" Proof can be shown by turning the process around and multiplying.

```
SQRT 100
RESULT: 10
PRINT 10 * 10
100

SQRT 1600394
RESULT: 1265.06
PRINT 1265.06 * 1265.06
1.60037E6
```

The latter example may raise questions about significant digits, rounding off, error, and scientific notation. These topics are often bypassed; Logo creates a surprising result which can serve as a real situation for exploring these ideas. Why doesn't the exact answer come back? What is close enough? Students usually think that mathematics deals only with exact answers. The calculator and Logo can be used to build understandings of arithmetic accuracy.

THE ROOT OF THE SQUARE

The square root function can be used to discover the length of unknown sides of right triangles. The length of the side opposite the right angle is equal to the square root of the sum of the square of side one and the square of side two. This length can be figured with a short procedure using OUTPUT.

```
     TO SIDE3 :SIDE1 :SIDE2
        OUTPUT SQRT ( :SIDE1 * :SIDE1 +
:SIDE2 * :SIDE2 )
        END
```

An understanding of arithmetic operations with variables may be needed. (For an exploration into the order of arithmetic operations, see "Tipps for Teachers," NLX October 1983, "The Truth about Numbers.") The operation OUTPUT needs to be introduced as a way of passing numbers or results around. Because OUTPUT is the action of SIDE3, it (with its inputs) becomes an operation and could be used as an input to the command FORWARD. Using a procedure as input may be a difficult idea which needs exploration.

```
SIDE3 60 80
RESULT: 100
PRINT SIDE3 60 80
100
FORWARD SIDE3 60 80
```

ONE MORE TRIANGLE

A triangle with two equal sides also has two equal angles. A triangle with two 45 degree angles is the easiest to draw using SIDE3. A description of a procedure for a 45 degree triangle might be:

```
To draw a right triangle with two 45
degree angles
   Move forward the length of the side
   Turn 90 degrees to the right (left)
   Move forward the same length
   Turn right (left) 90 plus 45 degrees
   Move forward the length of SIDE3
   Turn right (left) 90 plus 45 degrees
```

The sides are of equal length, so only one variable is needed.

```
TO 45.TRIANGLE :SIDE
   FD :SIDE
   RT 90
   FD :SIDE
   RT 135
   FD SIDE3 :SIDE :SIDE
   RT 135
   END
```

45.TRIANGLE is an independent module which can be used anywhere on the screen. Calculating SIDE3 with Pythagoras' formula makes this possible. New PYRAMIDS can be designed with 45.TRIANGLE. Houses can have other kinds of roofs. New SPIN.TRIANGLEs can combine 45.TRIANGLE and procedures for equilateral triangles.

As handy as 45.TRIANGLE is, it is still not an all-round, anywhere, anytime, any size, any angle triangle procedure. Next month, activities for a generalized triangle procedure will be developed as a Type III Problem (See "Problem Spaces in a Project Oriented Logo Environment" elsewhere in this issue) for Logo students in middle and high school.

Steve Tipps is a professor in the University of Virginia's School of Education, and presents Logo workshops for school systems throughout the United States.

Logo on Sale

Here are some low Logo prices. Krell Logo ($72.95) and DR Logo ($79.95), available from Nonagon Software, 11185 San Pablo Avenue, El Cerrito, CA 94530, (800) 942-2058. Terrapin Logo ($109.95), available from Ark Computing, PO Box 2025, Corona, CA 91720, (714) 735-2250. Also, Terrapin Logo ($112.46), available from Strictly Software, Inc., 28 Barrett Road, Katonah, NY 10536. (914) 232-4844.

MICROWORLDS

by
Glen Bull

Adventure Stories and English Skills

Involving a class in an interactive adventure story is an exciting way to work on written language skills. Members of the class write different segments of the story. At the end of each segment of the story, the reader gets to choose the direction the story takes. This idea has been used with success by Nancy Sopp and others to develop English skills in the classroom.

```
                JONATHAN'S ADVENTURE

JONATHAN PULLED BACK THE COVERS AND
LISTENED CAREFULLY. THE CRACKLING
NOISE APPEARED AGAIN. HE HOPPED OUT OF
BED AND RACED OVER TO THE WINDOW.
A LUMINOUS GLOW WAS VISIBLE THROUGH THE
LEAVES OF THE OAK TREE IN THE FRONT YARD.

WHAT SHOULD JONATHAN DO?

TURN ON THE LIGHT? (TYPE A)
WAKE HIS PARENTS? (TYPE B)
```

The most straightforward way to create a procedure which will print on the screen the text shown above is through a series of PRINT statements. This method is easily comprehended by students, and will work satisfactorily.

```
TO PART1
PRINT [ JONATHAN'S ADVENTURE ]
PRINT [ ]
PRINT [JONATHAN PULLED BACK THE COVERS AND]
PRINT [LISTENED CAREFULLY. HE HOPPED OUT OF]
etc.
```

Note: The procedures which follow are written in Terrapin Logo. A complete listing of the LCSI (Apple) version is provided at the end of the column.

This method requires some experimentation to determine the length of each line. I found that I was not patient enough to finish my story. As an alternative, the story could be written with the Logo editor, and could be printed out as a procedure.

```
TO PART1
   JONATHAN'S ADVENTURE
   JONATHAN PULLED BACK THE COVERS AND
   LISTENED CAREFULLLY. THE CRACKLING
   NOISE APPEARED AGAIN. HE HOPPED OUT OF
 etc.

 TO PRINT.STORY
    PRINTOUT PART1
    MAKE "CHOICE READCHARACTER
    IF :CHOICE = "A PRINTOUT PART2A
    IF :CHOICE = "B PRINTOUT PART2B
    END
```

I found that this solution was unsatisfactory because the procedure name TO PART1 appears at the beginning of the procedure printed out. The word END also appears at the end of the procedure. By this time, the prospect of using the Logo editor to write the story was tantalizing. I simply couldn't face the prospect of all those PRINT statements with a perfectly good Logo editor available.

Logo has a TEXT command available, which enables one to print the text of the procedure without printing the procedure name. However, it does not appear in the form I desire. Instead, each line of the procedure is enclosed in brackets.

```
PRINT TEXT "PART1

[ ] [ JONATHAN'S ADVENTURE ] [ JONATHAN
PULLED BACK THE COVERS AND ] [ LISTENED
CAREFULLY. THE CRACKLING ] [ NOISE APPEARED
AGAIN. HE HOPPED OUT OF ] etc.
```

Those who followed last month's column may recognize that each line is a list. A list in Logo is defined as a series of items enclosed in brackets. Logo procedures seem to be stored in the form of a series of lists.

It is possible to avoid printing the blank brackets (empty list) at the beginning of the procedure by using the Logo command BUTFIRST. The empty brackets are associated with the title line of the procedure, and usually contain variable input names, if any are present.

```
PRINT BUTFIRST TEXT "PART1

[ JONATHAN'S ADVENTURE ] [ JONATHAN PULLE
D BACK THE COVERS AND ] [ LISTENED CAREFUL
LY. THE CRACKLING ] [ NOISE APPEARED AGAIN
. HE HOPPED OUT OF ] etc.
```

The results were a little better, but still nothing that I would write home to Mom about. The strategy needed to print the series of lists is similar to that used for the report generator in last month's Logo grade book.

An English description of this process might be:

```
To write a story
   Print the first line in the story
   Remove the first line from the story
   Continue this process until all
      lines have been printed
```

The same idea could be expressed in Logo fairly easily. The procedure WRITE simply bypasses the empty set of brackets in the TEXT of whatever procedure we are concerned with. WRITE.LINE performs the remaining operations described above.

```
TO WRITE :STORY
   WRITE.LINE BUTFIRST TEXT :STORY
   END

TO WRITE.LINE :STORY
   IF :STORY = [ ] STOP
   PRINT FIRST :STORY
   WRITE.LINE BUTFIRST :STORY
   END
```

These two procedures now allow me to write the text of a procedure just as it appears, without the procedure name.

```
WRITE "PART1

JONATHAN'S ADVENTURE
JONATHAN PULLED BACK THE COVERS AND
LISTENED CAREFULLY. THE CRACKLING
NOISE APPEARED AGAIN. HE HOPPED OUT OF
etc.
```

My wants are simple. I would be perfectly content if the title of the story were only centered. Typing spaces before the title doesn't work, since the Logo editor removes extra spaces to conserve the amount of memory used.

Last month, during the development of the report generator, we found that printing character number 32 produces a space. This enables us to write a procedure which will print spaces.

```
TO SPACE :TIMES
   REPEAT :TIMES [ PRINT1 CHAR 32 ]
   END
```

If this procedure could be placed in the middle of the story, it could be used to move the title line of the story over to the center. The key question is, "How will the program recognize that this is a command, and not text to be printed?"

There are several ways in which this problem might be solved. One way is described in an article by Tom Lough and Steve Tipps in the special Logo issue of Classroom Computer News (April 1983, p. 53). The way in which I chose to solve the problem was by placing commands in brackets in the Logo editor. If text is enclosed in brackets, I assume that it is a command, and execute the command. If text is not enclosed in brackets, it is printed.

For example, the command, BLANK.LINE, could be added to the story to place a blank line between the title and the body of the story.

```
TO BLANK.LINE
   PRINT [ ]
   END
```

The story would then look like this.

```
TO PART1
  [ SPACE 9 ]
  JONATHAN'S ADVENTURE
  [ BLANK.LINE ]
  JONATHAN PULLED BACK THE COVERS AND
  LISTENED CAREFULLY. THE CRACKLING
  NOISE APPEARED AGAIN. HE HOPPED OUT OF
  etc.
```

But we're not finished yet. First, we need to tell Logo how to decide whether to print or execute a line. The Logo instruction called LIST? (LISTP in Apple Logo) can be used to check whether text is enclosed in brackets.

```
LIST? [ DOG ]
RESULT: TRUE
LIST? "DOG
RESULT: FALSE
```

Thus, LIST? indicates that a line is a command because I have enclosed my commands in brackets. This method of discriminating between commands and text can be easily incorporated into WRITE.LINE.

more Microworlds

```
TO WRITE.LINE :STORY
   IF :STORY = [ ] STOP
   TEST LIST? FIRST FIRST :STORY
   IFTRUE RUN FIRST FIRST :STORY
   IFFALSE PRINT FIRST :STORY
   WRITE.LINE BUTFIRST :STORY
   END
```

The result of the WRITE procedure is shown below.

```
WRITE "PART1

       JONATHAN'S ADVENTURE

JONATHAN PULLED BACK THE COVERS AND
LISTENED CAREFULLY. THE CRACKLING
NOISE APPEARED AGAIN. HE HOPPED OUT OF
etc.
```

This line by line method works best if <return> is pressed after each line in the story is entered in the editor. The revised method of piecing together the adventure could have the following appearance.

```
TO TALE
   CLEARTEXT CURSOR 0 0
   WRITE "PART1
   MAKE "CHOICE READCHARACTER
   IF :CHOICE = "A WRITE "PART2A
   IF :CHOICE = "B WRITE "PART2B
   END
```

This is the WRITE stuff for printing adventures. An extension of this method could also be used to generate letters to parents and end of school year reports.

LCSI (Apple) version

```
TO TALE
   CLEARTEXT SETCURSOR [ 0 0 ]
   WRITE "PART1
   MAKE "CHOICE READCHAR
   IF :CHOICE = "A [ WRITE "PART2A ]
   IF :CHOICE = "B [ WRITE "PART2B ]
   END

TO WRITE :STORY
   WRITE.LINE BUTFIRST TEXT :STORY
   END

TO WRITE.LINE :STORY
   IF EMPTYP :STORY [STOP]
   TEST LISTP FIRST FIRST :STORY
   IFTRUE [RUN FIRST FIRST :STORY]
   IFFALSE [PRINT FIRST :STORY]
   WRITE.LINE BUTFIRST :STORY
   END

TO SPACE :TIMES
   REPEAT :TIMES [ TYPE CHAR 32 ]
   END
```

BLANK.LINE and PART1 are the same as in the Terrapin version.

Glen Bull is a professor in the University of Virginia's School of Education, and teaches Logo courses at both the graduate and undergraduate level.

Computer Game Contest

Scholastic Inc. and Verbatim Corporation announce the Verbatim Computer EdGame Challenge, a contest for classes, individual students, teachers, and professionals to design and write educational computer games. Hardware prizes totalling over $12,000 will be awarded to the winners. Only non-violent games will be judged. The contest closes on April 30, 1984. Here is a chance to show what Logo can do! Write for your entry blank and rules today. Send your request to Corona B. Gray-Moore, Scholastic Inc., 730 Broadway, New York, NY 10003, or call her collect at (212) 505-3485.

Children Love Turtle Puppets

Have you been looking for a turtle puppet? Cynthia Solomon of Atari Research told us about the Folktails Turtle from Folkmanis. It has the highest quality of any turtle hand puppet we have seen. Handcrafted by conscientious workers and measuring about 12 inches long and 5 inches high, the puppet comes with care and handling instructions and a short story on how the turtle got his shell. You can make the turtle pull feet and head into its shell and pop them out again. This puppet will find many uses in Logo classrooms. Check in your local high quality toy store. In case you cannot find the puppet, a store in Charlottesville which stocks the puppet has agreed to make them available by mail to NLX subscribers. Send a check for $16 to Persimmon Corner, Folktails Turtle, 421 East Main Street, Charlottesville, VA 22901. VA residents enclose 64 cents for state sales tax.

Of Sushi and Mushi

by

Duane Yee

In an October 9, 1983, article in Parade magazine entitled, "Can We Rise to the Japanese Challenge?", David Halberstam tells about the "sobering experience" of a trip to Japan in which he observes so much that is superior to America, both in industry and in industriousness.

One of the photos accompanying the article depicts primary-age Japanese children seated before computer keyboards and monitors, with the caption, "Computer learning starts early: A kindergarten class in basic mathematics." Yet, about the only reference in the article to Japanese elementary education in any form is a single sentence stating that the "Japanese have emphasized science and engineering in their schools ..."

Mr. Halberstam or the editors of Parade have fallen victim to America's paranoia about Japanese competition. They have incorrectly assumed that a country so abounding in high tech expertise must also be a leader in computer education in the classroom. In a July meeting in Tokyo, I found that, at present, computer education is virtually non-existent in the traditional Japanese schooling.

GETTING TO KNOW EACH OTHER

My hosts were called the Society for Studying Logo in Education - a small consortium of educators, hardware manufacturers and software people. The corresponding secretary of this fascinating group of computer enthusiasts had the most appropriate surname of Goto. Mr. Hiroyoshi Goto is a member of the Electronics Division of the Uny Company, a huge supermarket chain based in Nagoya, and developer of a Logo version for the Hitachi Model 5 computer.

I had been put in touch with the society by Tom Lough, editor of the NLX, who, upon hearing that I was to vacation in Japan, felt that I might need some time off from sampling sushi to engage in more rigorous mental activity. The result was my three-hour meeting at Tokyo University with people from Kyoto, Nagoya, and Tokyo. (Not coincidentally, Tom had written an article in the April 1983 issue of Electronic Learning magazine about our work with Logo at Punahou School.)

One of the society's major objectives is to promote computer education in Japan, be it through traditional schools, primary schools (preschools and kindergartens), or through the large network of "juku" - tutoring schools of various sorts that meet with students after school and on non-school days.

The takeoff point for our discussion was a slide show I presented on Punahou's Logo program. There was such interest expressed in the Punahou slides that they were used later in two national Japanese conferences, one involving primary teachers, and the other educators in general.

LOGO FOR JAPAN

The Punahou Logo student activity cards we had developed caused great excitement, to the extent that the Uny Company sent two representatives to Scott, Foresman & Co. in Chicago - the publishers of the cards - the first week in September to negotiate for translating the learning materials into Japanese.

When the Uny Company's representatives stopped by to visit Punahou on their return to Japan, they indicated that the society's Japanese translation of Logo was expected to be completed by the end of September.

I also learned of the Fujitsu company's plan to market its own machine specific translation of Logo.

Several members of the society were engaged in translating several different works on Logo, including Abelson and diSessa's Turtle Geometry (Ed. Note: see NLX Vol 1 # 2, October 1982 for review), and some writings on Logo in special education. Time did not permit me to find out about the special "mushi" (Japanese bugs) encountered by those brave souls who were doing the translation!

I also had the opportunity to learn about the new Yokohama Children's Science Museum which is being planned for completion by the end of 1984. It is patterned on the exploratorium model with lots of hands on

Sushi continued

experiences, and, of course, will include microcomputers and Logo. I learned also of one of the first children's computer camps in Japan - on Shikoku Island in August, sponsored by the Primary Schools Association. Mr. Goto was one of its principal architects, and it featured students from primary school age through high school.

READY TO GO

So I found a Japanese educational system that was far from being replete with computer education. But I also found a group of people - educators as well as non-educators - who were intensely interested in promoting computers in education, both traditional and non-traditional.

As for the "Japanese Challenge" suggested by Mr. Halberstam and Parade magazine, I can only respond by citing my own experiences. It is somewhat dubious that even the most optimistic of my Japanese friends would envision an intrusion of computers into the traditional school system in the immediate future.

But how long is immediate? And, if success with computers and children is indeed coming to the fore, will not even the most conservative traditionalists look towards computer education in all schools? One might consequently expect an incredible burgeoning of computers in Japanese education - but they're not there yet.

Duane Yee is the principal of the Punahou Junior School (K-8) in Honolulu, HI.

SPRITES!

Synetix Inc. announces a series of sprite boards for the Apple computer. They have 32 sprites, all of which can move simultaneously. Background scenes can be created with devices (paddles, joysticks, etc.) connected to the game port. Sprite drivers will be available for MIT Logo (Terrapin, Krell) and Apple Logo by February, according to spokesperson Dave Boyd. Other features include a sound generator and speech synthesizer. These may not operate with Logo, however. Prices range from $149 to $395. For more information, write Synetix Inc., 10635 NE 38th Place, Kirkland WA 98033, or call (800) 426-7412.

NLXual Challenges

with Robs Muir

Perhaps the pleasure of problem inquiry in Logo is its resemblance to play. The goal is sometimes ill defined and, often, non-utilitarian.

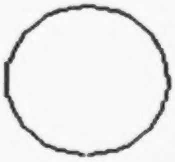
The beauty of Logo problem solving is that the limits are those of our imaginations. We have fun if we can meet our self-imposed tasks. We find joy along the way if we discover some fundamental truths about mathematics, computers, or the cognitive processes.

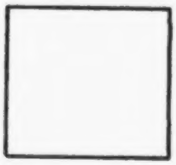
We define the goals, and we set the point at which our play is concluded: our own rules, our own game. One such game involves the turtle geometry of circles.

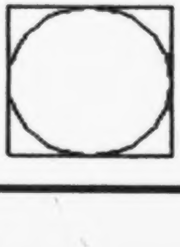
Have you ever struggled with drawing different sized circles using the REPEAT command? The idea of a circle being composed of straight lines is odd; the idea of repetition complicates the problem even more.

Since we adults have all mastered the circle (haven't we??), perhaps we can move on to a problem which uses some "circular logic." This month's NLXual Challenge is a problem with two simple elements, the square and the circle. Perhaps a square can help us discover something new about Logo's circular polygons.

Take a simple shape, like a square, and construct a circle within its boundaries. The trick is to make the circle tangential to each of the four sides, something like this:

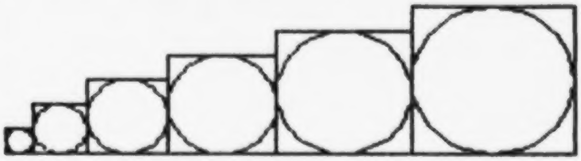






Challenges continued

Doesn't this sound easy? (Oh, no, you Apple Logo people! No fair using the pseudo primitives CIRCLER, CIRCLEL, ARCR, or ARCL from your STARTUP file!) Once you have a solution, let's take it one step further. Write a procedure which will make this shape any size you'd like, i.e., a square of 37.65 turtle steps per side. How succinctly can you write your procedure(s)?



Another step? OK. Let's draw circles in the center of other polygons: triangles, pentagons, hexagons, etc. In fact, what is the limit at which our polygon is apparently indistinguishable from the circle inside it? (Use a standard diameter of, say, 100 for your circle.)

Your move ...

Note: As with all NLXual Challenges, if you want to receive a possible solution, send a copy of your solution (or your attempt) with a stamped self addressed envelope to NLXual Challenges, Attn: Robs Muir, 1688 Denver Avenue, Claremont CA 91711. We will compile contributions and send you the most elegant solution.

If you would like to submit a challenge or puzzle for this column, send your ideas to the above address.

Robs Muir is a teacher in the Claremont CA Unified School District and Claremont Graduate School. He presents Logo workshops in the southern California area.

News

The Quadlink board for the IBM PC will run Apple Logo (LCSI), according to a preliminary list published by the Quadram Corporation. No other Logo versions for the Apple computer were listed. For more information, write Quadram Corporation, 4355 International Boulevard, Norcross, GA 30093, or call (404) 923-6666.

American Indian Systems Co., Inc. is holding a three day workshop for Indian educators entitled, "Microcomputers within Indian Education," February 28 through March 1 at the El Dorado Hotel in Reno, NV. The workshop will include presentations on Logo and its use at summer camps for American Indian children. For more conference information, write AISC, PO Box 214844, Sacramento, CA 95821. If you are aware of Logo use within the American Indian population, please contact Evelyn Dale, 820 Christie Court, Davis CA 95616.

Micro, a new magazine from Japan, features Logo in many places in its premier issue. Although written mostly in Japanese, there are several English words (such as Papert, Logo, and PENUP) in the articles. On page 58 is a news article about Duane Yee and the Logo Activity Cards developed by his teachers at Punahou School in Honolulu. (Ed. Note: also see Electronic Learning, April 1983, and Duane Yee's article in this NLX issue.) A Japanese magazine for Logo users, entitled, Logo Some Z's, is introduced. Several Logo versions (MIT, LCSI, Commodore 64 Logo, and Japan's H/Logo) are discussed. A photo of Terrapin's Mark Eckerwiler also appears. For more information on Micro, write to Akihiro Kato, Micro Magazine, Publishing Division, Shinkigensha Inc., Shinjuku-Youth Building 1-9, Shinjuku 4 Chome, Shinjuku-Ku, Tokyo 160, Japan, or (if you can afford it) call 03-350-0848.

The National Logo Exchange, copyright 1984 by Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly, September through May, $25 (US) per year, mailed FIRST CLASS from Charlottesville, Va. $5 (US) additional per year for addresses outside the USA, Canada, and Mexico. The opinions expressed by the authors are not necessarily those of The National Logo Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress Street, Salem, MA 01970. Address other correspondence to: The National Logo Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor..Tom Lough

ISSN 0734-1717

The National LOGO Exchange

Volume 2 Number 6 FORWARD 100! February 1984

Concrete Turtling

by Joan Rainey

Whilst on leave in Boston for a year, I volunteered to teach Logo to a small group of students at the Lexington Montessori School. During that time, the importance of the concrete level of learning was impressed upon me anew.

The students ranged from grades 1 to 3, and had no previous Logo experience, and no typing ability. I chose as my basic objective to introduce the students to programming through the Logo language, within the intellectual and mathematical ability of each child.

After several experiments with groupings, I decided to concentrate on individual tutoring. The resulting months became one of the most challenging and fascinating educational experiences I have had in my entire teaching career.

As my knowledge of Logo was developing at the same time as that of the children, I learned with them, and from them. With my background in the use of the Baratta Lorton Method to introduce elementary mathematics, I became excited at the potential for curriculum enrichment as well as learning to program in this unique language.

I concentrated on three main areas of learning:

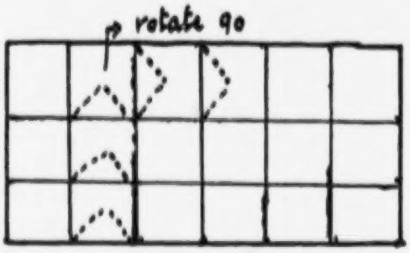
1. understanding the grid system
2. developing concepts of angles and sides in polygons
3. developing pattern block designs

In these areas, I made use of concrete materials, working at the concrete level before abstracting, writing programs on paper before reproducing them symbolically on the computer. This resulted in a three way process of concrete level - connecting level - symbolic level.

GRIDS

To facilitate the understanding of the grid system, I developed my own method, using the "Mathematics Their Way" concept of tiles. The children built their own grid system by setting out tiles on a table or floor.

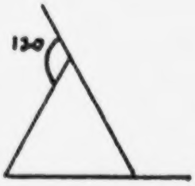
They plotted their own designs by using small triangles, which they placed on the tiles to show the path of the turtle. This aided their understanding of directionality, angles, and the number of steps FORWARD and BACK.



From this, the children were able to create designs, write the programs to produce them, and transfer them to the computer. The extra time taken at this stage with concrete materials promoted a greater understanding of the grid system and the concepts under investigation.

OUTSIDE ANGLES

In order to introduce and reinforce the more complex geometric concept of outside angles, I decided again to work first at the concrete level. Using the Montessori sticks and other materials, we explored shapes and angles, then inside and outside angles. By discussion and exploration, we found out how far the turtle would have to turn to change direction.

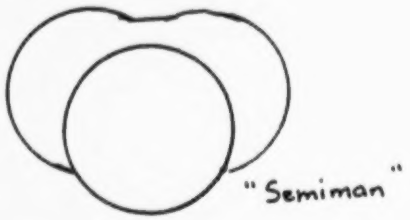


Concrete Turtling continued

A study of polygons followed. After drawing our shapes on the computer, we extended our knowledge by working out a procedure for each shape. Repeat designs were also introduced by changing the angle before drawing the shape again.

By now, the children could write complete programs for polygons, draw them on the computer, and then define a procedure for each shape. The procedures were used in extension activities.

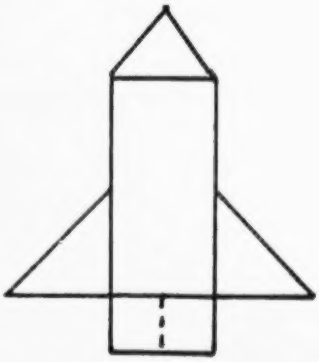
During this time, the concept of a circle being a many sided polygon was explored, first with the Montessori materials, and then with the computer. The importance of 360 was discovered and discussed. Strange circular creatures appeared on the computer screens.



PATTERN BLOCK DESIGNS

As reinforcement, pattern blocks were used to reproduce designs which were later transferred to the computer. I made a set of class cards for the children, incorporating the different shapes.

As a result, the children were making rocket ships, kites, and flowers out of a combination of shapes they had learned to draw. Modular thinking wasn't too far away.



IN SUM

I feel that the strategy of using concrete materials results in a greater understanding of the relationships and symbols. The children have an opportunity to do something, feel something, see and touch something. They have the opportunity to learn something other than by being shown.

Being able to express a relationship in terms with which they are familiar became less mysterious. Additionally, the concrete operations allowed children to be more involved in the process.

I had to continually remind myself, however, that each child is an individual, and has his own way of developing a learning style. Each child's mind and the "mind sized bites" will be based on that individuality. I believe concrete turtling promotes this process.

Joan Rainey has taught at the primary and junior level in Northern Ireland and Canada. She is teaching presently in Kingston, Ontario.

Infoworld Logo

"Where is Logo Taking our Kids?" is the cover story of Infoworld (January 23, 1984) by Scott Mays. He reports that Logo is taking the educational system by storm and may be poised to "replace BASIC as a language for first time users." Jim McCauley and John Allen are quoted on their perspectives about use of Logo and its future. Mays also talks about features available in new versions and includes a comparison chart.

The follow-up article by Barbara Wierzbicki in the same issue states, "The success of Logo hinges on teacher training." Logo is a new educational philosophy which challenges teachers to redefine their roles, according to Robert Taylor of Columbia Teachers College. Several other interviews emphasize the necessity of having teachers overcome their fear of the equipment and experience the joy of learning with Logo. The potential for Logo as a way of individualizing education is great, but teachers must be well trained in the philosophy as well as the mechanics of the language.

From The Editor

As I was reading recently, I came across the statement, " ... and the Greek word KATALOGOS means to list." I thought to myself, "How fitting that a word containing 'Logo' also carries with it one of the major facets: lists!"

So often, what the public reads about Logo is a surface treatment. Articles containing information about turtle graphics abound. As a result, the popular conception has been formed that Logo is just a graphics language for young children.

Consider the following statement, taken from a report on a recent computer contest for teams of elementary school children. "We will probably never see entries submitted in Logo, since a team would have to be far more sophisticated with the Logo language than with BASIC to be able to solve the same problems."

As long as the list and computational sides of Logo are dismissed with such statements, I fear that Logo's potential for education may be misconstrued by those who make policy decisions.

We must make a concerted effort to inform our administrators, parents, and students that Logo is more than just a set of graphics commands. Unfortunately, that means we all must travel deeper into Logoland. We must be willing to grapple with the more difficult aspects of the language. We must help our students understand, appreciate, and benefit from the higher powers of Logo.

Newer Logo versions are becoming available for more powerful computers (such as the IBM PC), with capabilities of addressing large amounts of memory and performing mathematical calculations with up to 1000 significant figures and magnitudes up to ten to the 10,000th power! Additionally, file handling capabilities are improving rapidly.

With such changes coming, the training of teachers takes on an even more urgent priority. Although there will be a continuing need for beginner Logo classes, the training for more advanced uses of Logo in the classrooms should not be neglected.

If our educational systems can respond to this challenge successfully, then the full power of Logo can be effectively demonstrated to the public through the children themselves.

The suggestion made by an article in Time magazine that no one can get a job programming in Logo will soon be incorrect.

Until that time, FD 100!

Tom

Terrapin Announces Updated Logo for Apple Computers

Terrapin, Inc. has announced Version 2.0 of Terrapin Logo for Apple computers. The update adds 6 new primitives, including list operators COUNT, ITEM, EMPTY?, and MEMBER?. Several new file management and editor commands will also facilitate the advanced uses of the language. Recalling previous lines which include REPEAT using CONTROL-P is also a welcome improvement.

An especially helpful feature will be the ability to read program files created with LCSI's Apple Logo. This is a real advance for compatibility. Improved garbage collection capabilities will help with the limited workspace in the Apple memory.

An extensive chapter on "Words and Lists" has been produced as an addition to the documentation manual. Terrapin president J. Sheridan McClees said he is particularly pleased with the improvements made on the upper end of the language so that the advanced programmer will have many more possibilities.

The price for the complete new package will be $149.95. Updates for owners of Version 1.0 cost $20.00. The new "Words and Lists" chapter can be obtained for an additional $5.00 for previous owners.

For more information, contact Terrapin Inc., 380 Green Street, Cambridge, MA 02139, (617) 492-8816.

As Easy As 1,2,3

A Review by

Pat Stevens

It is erroneously called "A Logo Funbook for Kids." Donna Bearden's 1,2,3, My Computer and Me is for parents and teachers, too!

In this delightful book, the author draws upon her background of experience with the Young Peoples' Logo Association, and sets forth clear easy steps to learning how to tell the turtle what to do.

Young people of all ages will find how to test their hunches and explore mathematical ideas in a light hearted manner. Each chapter develops new concepts using discovery techniques. There are also a variety of related off computer activities, which we often find missing in other books.

Each step of the way, the author offers blank spaces for the student to fill in with results as new ideas are developed and explored. Instead of telling the reader that a right angle turn is 90 degrees, Bearden encourages a sequence of attempts to discover the relationship by trying different numbers and evaluating the results.

A typical page reads as follows:

"I tried these numbers:

___ was too ___
___ was too ___ ..."

Additionally, the author is sensitive to anxiety due to having a different Logo version than the particular one used in the text. Throughout the book, she provides all commands for Apple, MIT, and TI Logo.

After my class had been introduced to the idea of a variable, we used the following procedure from the book to explore the angles for five sided figures.

```
TO PS :N
   REPEAT 5 [FD 45 RT :N]
   END
```

As you can imagine, PS stands for pentagon, star. We started by trying to find the correct angle for the pentagon. We saw that it was less than a right angle. As it became clear that 60 was too small and 80 was too large, the excitement mounted. Some tried 70, 71, 74, 73 ... when we found that 72 was perfect, the children exploded with delight!

In a similar manner, we found the turn of 144 for a star. The children were thrilled. From these beginnings, we went on to write our own procedures for pentagons and stars of all varieties. I appreciate a book which helps bring such excitement to my classroom.

At the end of each chapter, the author presents ideas for experimenting further. In some cases, drawings are offered, for which the student is invited to discover a procedure. Occasionally, a hint is given.

(I spent over an hour working on one for which no hint was provided. I'm not sure if I would have preferred a hint to position the turtle at a critical point, or if my feeling of success at the end was worth the frustration along the way.)

The book continues with more variables, recursion, and x-y coordinates. One of the later chapters deals with using colors. Information on saving procedures was a little too sparse for me.

Although the book is written for elementary school students, it can be a source of fun, interest, and learning for Logo beginners of any age from 2 to 102!

1,2,3, My Computer and Me is published by Reston Publishing Co., 11480 Sunset Hills Road, Reston, VA 22090, (703) 437-8900. $10.95.

Pat Stevens is an elementary school teacher in Santa Fe, NM, and is actively involved in promoting the use of Logo throughout the southwest.

Logo on Sale

Oryx Systems is offering Krell Logo for $75, Logo for the Apple II (Abelson) for $15, and Terrapin Logo with full documentation for $115, according to a recent advertisement. (800) 826-1589. Be sure this is the latest Terrapin Logo release, Version 2.0.

TIPPS for TEACHERS

by
Steve Tipps

Triangle Thinking: SIDE 2

Triangle thinking in last month's column concentrated on making and maneuvering triangles. The dependable equilateral triangle with inside angles of 60 degrees (and outside turns of 120) was used to make pyramids.

Then, an automatic triangle procedure based at HOME created any size right triangle, but it was always fixed at one corner. Finally, the discovery of the Pythagorean theorem presented as a Logo exploration enabled students to make isosceles triangles which could be moved anywhere on the screen.

Exploration of triangles in Logo continues this month with two very different kinds of problems. Filling triangles provides an open exploration with some surprising results. Creating a moveable right triangle points to learning about relationships between the three sides.

FILLING TRIANGLES:
A TYPE I PROJECT

Drawing shapes is a beginning place in Turtle graphics, but outlining shapes are often inadequate for the needs of students' drawings. They often want filled in shapes: squares, triangles, and trapezoids. This desire is a marvelous basis for exploration of geometric properties of these shapes and creates a need for new or more refined Logo skills. Glen Bull and I call this a TYPE I project, in which children are initiators and the teacher is a facilitator.

Although the goal may be to fill a triangle, a filling strategy can be developed more easily for a square or rectangle at first. One way is to start one side, draw the base to the other side, turn 90, move up one or two lines, turn 90 again and draw back to the first side. Drawing this process first on paper with a magic marker will help determine the distance and directions of the move. The sequence is repeated until the box is filled.

```
TO FILL.TWO.LINES
   RIGHT 90
   FORWARD 50
   LEFT 90 FD 1 LEFT 90
   FORWARD 50
   RIGHT 90 FD 1
   END

TO FILL.SQUARE
   REPEAT 25 [ FILL.TWO.LINES ]
   END
```

This strategy is obviously less flexible than one using variables and is less elegant than one using recursion. The variable change is fairly straightforward.

```
TO FILL.TWO.LINES :SIDE
   RIGHT 90
   FORWARD :SIDE
   LEFT 90 FD 1 LEFT 90
   FORWARD :SIDE
   RIGHT 90 FD 1
   END

TO FILL.SQUARE :SIDE
   REPEAT :SIDE/2 [FILL.TWO.LINES
:SIDE]
   END
```

My first attempt at a recursive square follows. The logic was to use the side as the dimension to test and also as the variable to decrement in the recursive line.

```
TO RECURSIVE.FILL.SQUARE :SIDE
   TEST :SIDE <1
   IFTRUE STOP
   FILL.TWO.LINES :SIDE
   RECURSIVE.FILL.SQUARE :SIDE-2
   END
```

When I tested this procedure, I was surprised (and delighted) to find a bug in my thinking. Instead of a filled square, I got a recursive fill triangle! Upon closer examination, I saw that the relative sizes of the sides can be controlled by the size of the decrement. A :SIDE-2 gives an equilateral triangle with two sides of size :SIZE. A :SIDE-1 gives a right triangle with one side twice as long as the other. A :SIDE-4 makes the ratio of the sides 4 : 1.

Since FILL.TWO.LINES fills two lines of the same length, the slope of the third side may not be accurate enough for you. But with a little finagling, FILL.TWO.LINES can be divided into two parts in which the return line length is reduced separately. That is

Tipps continued

left as an exercise. However, the fortuitous RECURSIVE.FILL.TRIANGLE has left the problem of RECURSIVE.FILL.SQUARE unsolved.

Perhaps I needed a way to keep up with two different elements: the side of the square and the dimension to be decremented. With a square, the beginning size of both is the same, so I could use two different variables of equal size.

```
     TO RECURSIVE.FILL.SQUARE :SIDE
:DUMMY.SIDE
        TEST :DUMMY.SIDE <1
        IFTRUE STOP
        FILL.TWO.LINES :SIDE
        RECURSIVE.FILL.SQUARE :SIDE
:DUMMY.SIDE-2
        END
```

Try the following:

```
RECURSIVE.FILL.SQUARE  50 50
RECURSIVE.FILL.SQUARE 100 30
RECURSIVE.FILL.SQUARE  20 127
```

With :SIDE and :DUMMY.SIDE equal, I made a very adequate square. Then I tried other combinations and found that I had an all purpose rectangular filler. This led to a new procedure with two variables, :SIDE1 and :SIDE2.

```
TO RECURSIVE.FILL.FRAME :SIDE1 :SIDE2
   TEST :SIDE2 <1
   IFTRUE STOP
   FILL.TWO.LINES :SIDE1
   RECURSIVE.FILL.SQUARE :SIDE2-2
   END
```

This family of filled-in shapes might be completed with a circle (See NLXual Challenges, elsewhere in this issue. Ed.), hexagon, trapezoid, etc. Developing the procedures is an excellent classroom project with many strategies possible.

WHY FIDDLE WITH FILLING?

Although the goal was to design a filled in triangle, the discussion of filling has not stayed specifically with triangular thinking. In such a TYPE I problem or project (See NLX, Dec 83 and Jan 84), many outcomes are possible. Although the original problem focused on triangles, the solution opened many other possibilities. The mistakes and side trips which I tried to report were as valuable to me as the procedure RECURSIVE.FILL.TRIANGLE, the purported intention.

The description is not a lesson plan for filling shapes, but a preparation to help the teacher ask the right question or make the right suggestion. Working through the problem ahead of time for the teacher is like travelling the path to know what pitfalls and potholes may lie ahead.

In TYPE I problems, however, the teacher does not announce them all ahead of time. Students are free to fall in some of the same holes and get out by themselves. The teacher can move to prevent total discouragement and rejection of the problem. Working through the problem also gives the teachers an appreciation of learning from their own mistakes.

Open ended exploration of filling has several other aspects worth noting. First, filling can be done successfully at different levels of Logo skill. A square can be filled through the rather laborious job of programming each line separately. REPEAT is a "better" (more parsimonious) programming technique, and recursion is even more refined. However, at each level of skill, the description of what the turtle is to do is the same.

Filling provides a natural situation for enhancing Logo skills. I have found that filling has been the first project in Logo which many students encounter that cries out for recursion. While recursion certainly can be "taught," Logo teachers look for situations in which needs for skills are almost spontaneous. Filling seems to be one of those situations.

FILLING DIFFICULTIES

There are problems with filling. When filled shapes on the screen overlap, distortion is frequent. Picture elements tend to capture each other and change their colors. This is a particular problem when drawing off axis.

Second, stopping in the middle of a project which demands a fill to do this exploration is a problem for some students. They may become discouraged when they are intent on one project and have a big delay to work out how to fill an area. Many would rather have the "right" procedure handed to them than find out how it actually works.

more Tipps

The teacher has to strike a balance of helping projects along and helping children understand how procedures work so that students can make them work. Sharon Burrowes makes that point well in her article "The Hazards of Hacking," in last month's NLX.

Finally, there are other ways of filling: light pens, Koala pads, or the FILL command in IBM Logo. The question is whether the students should use these other techniques or whether they should go through the process of finding out how they can do the same thing for themselves.

When the other techniques are widely available, FILL may become a purely academic exercise which no longer has a natural place; but if that comes to pass, I will mourn its loss as a stimulating TYPE I problem.

THE RIGHT TRIANGLES MOVE AWAY FROM HOME: A TYPE III PROBLEM

A different approach to learning with Logo is a TYPE III problem, characterized by the teacher with the intention of using Logo to explore content. The goal is not Logo per se, although a lot of Logo is learned in the process. Instead, the goal is to discover ideas using Logo.

One such problem would be an extension of the triangular thinking presented in last month's column. The AUTO.TRIANGLE procedure draws any right triangle, using HOME as a starting and stopping place. This anchor allowed students to discover the relationships between the two given legs and the length of the hypotenuse, using the Pythagorean theorem. In that respect, the column last month was a TYPE III problem solution.

The purpose of the problem is to discover the relationships among the three sides. Those of us who have forgotten, (or never understood) the definition of sine, cosine, and arctangent are in for a refresher course in trigonometry.

This month, the problem is set up for analysis. I invite you and your students to embark on the problem to see what pitfalls and potholes you discover. The estimated level of the problem would be seventh-eighth grade through college and adult.

*_

HOW ARE WE RELATED?

Conversation between three sides of a right triangle:

SIDE1 to SIDE2: Let's get together and tell SIDE3 how big to be.

SIDE2 to SIDE1: That's right, mate. If I make a square and you make a square, the sum of squares determines how big SIDE3 is going to be. (That is, if SIDE3 is smart enough to take the square root.) We make a pretty good team.

SIDE3: Hold on one minute. You are forgetting about our angles. I can work with either of you two sides to figure out what size our angles are. I think if we get that straight, we can all move away from HOME. I'd really like to run away from HOME and draw on the left side.

SIDE2: I've been wanting to go South of HOME. I bet when you can add us together, or maybe multiply or divide us, something interesting happens.

SIDE1: I'll bet our squares are important. Maybe we are all cousins.

SIDE3: We may have to try out many different things to figure out how we are related. Let's change our lengths and do a lot of calculations to see what we can find out.

*_

GOAL: To derive the relationships between length of sides and the angles in right triangles of any size.

SITUATION: AUTO.TRIANGLE can be used to generate triangles of any size using HOME as an anchoring spot. However, these triangles cannot be transported around the screen to use in other ways. If a relationship between the sides and angles could be found, a generalized procedure for ANY.TRIANGLE could be written with inputs of :SIDE1 and :SIDE2.

HINTS: The Pythagorean theorem is needed to determine the size of :SIDE3. Students should be able to use Logo to figure :SIDE3 and to construct 45 - degree isosceles triangles of any size. (See "Tipps for Teachers", Triangle Thinking: Side 1," NLX, Jan 84.)

Tipps . . .

HEADING, SETHEADING, and SETHEADING TOWARDS is useful for finding out what direction the turtle is pointing. The Total Turtle Trip Theorem can be used to change headings into degrees. PRINT out the amount of the turns and the length of the sides to help see the relationships.

```
PROGRAM STUB:
        PRINT HEADING - 90
        PRINT 360 - 90 - (HEADING - 90)
        or
        PRINT 360 - HEADING
```

RESULT: ANY.RT.TRIANGLE :SIDE1 :SIDE2 could be drawn at any place and any orientation on the screen.

This is an attempt to model a TYPE III problem. Your comments and solutions, alternatives and refinements, are invited. Some possible solutions will be discussed next month.

Steve Tipps is a professor in the University of Virginia's School of Education, and presents Logo workshops for school systems throughout the United States.

Have A Logo Summer

Dan and Molly Watt, and Tony Stavely will host The Logo Institute at Keene State College, July 30 through August 17. It promises to be an outstanding program, with many different activities planned. Included will be courses on Logo philosophy, list processing tools, and interactive programming. For more information, contact Dr. Howard Croteau, Director of Continuing Education, Keene State College, Keene, NH 03431, or call (603) 352-1909.

Michigan State University will offer a seminar on Logo research and educational applications, beginning June 20. Students will survey the theoretical models and educational applications of Logo, and become familiar with data processing aspects of the language for three hours of graduate credit. For more information, contact Dr. Lawrence T. Alexander, CEPSE, Michigan State University, East Lansing, MI 48824-1034.

The Department of Artificial Intelligence of the University of Edinburgh will teach both an elementary and an advanced Logo course this summer, beginning in mid-August. The elementary course has no prerequisites, while the advanced course requires at least one year of Logo experience. Advanced topics include list processing and using Logo in language teaching. The cost will be approximately $200. For more information, write K. R. Johnson, Department of Artificial Intelligence, Forrest Hill, Edinburgh, Scotland EH1 2QL, or telephone 031-667 1011, extension 2497. Please enclose an international air mail postal coupon with your inquiry.

Fresno Pacific College's immensely successful Summer Computer Institutes will be offered again, beginning in June. Arthur Wiebe, program director, reports that Logo became the language among the more than 500 attendees last summer. This year, each institute will have 6 laboratories, with 20 computers in each. A minimum of 50 hours of hands-on instruction is scheduled for each participant, for 4 hours of graduate credit. The fee is $350. Early reservations are encouraged, because space is limited in this popular offering. For more detailed information, write to Dr. Arthur Wiebe, Division of Education, Fresno Pacific College, 1717 South Chestnut Avenue, Fresno, CA 93702, (209) 251-7194.

If you have been looking for a Logo course to help you integrate the language into your regular curriculum, then consider a summer course offered by the University of Maine. "The Computer and the Curriculum: Learning with Logo," is designed for elementary and middle school teachers, administrators, and curriculum developers and will include both Logo study and Logo use in curriculum areas such as reading, mathematics, social studies, and science. No previous computer experience is necessary. Three credit hours are offered to students completing the course. For more information, write Dr. John W. Butzow, Department of Education, University of Maine, Orono, ME 04469, or call (207) 581-2434.

A workshop for Logo beginners will be offered by Jim Fejfar in Lincoln, NE, May 21 to June 8. Hands-on Logo work will be emphasized. Enrollment is limited. Write to Dr. Jim Fejfar, 214C Henzlik Hall, Lincoln, NE, 68588-0355.

MICROWORLDS

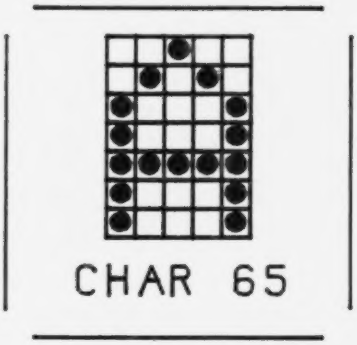
by
Glen Bull

Logo Characters

The TI 99/4 was the first computer with Logo to which I had access. It had an abysmal keyboard, but it also possessed a 16-bit microprocessor and a powerful Texas Instruments graphics chip.

TILES AND CHARACTERS

Tiles were a prominent feature of TI Logo. A tile consisted of a grid of squares which could be darkened to form shapes. TI Logo provided a set of ready-made shapes, which included the alphabet. For example, an "A" might be formed from a five-by-seven matrix of darkened squares.


CHAR 65

The alphabetic characters produced by most computers are formed similarly by dots in a grid. If you have a monochrome monitor (rather than a television screen) it may be possible to see the individual squares which form the letters.

The characters of the TI alphabet could be altered by changing the pattern on the tile. Foreign alphabetic symbols could be substituted for the English alphabet, for example. The recently released IBM Logo has similar capabilities.

I am currently working on a Logo program to produce tiles on the IBM Personal Computer similar to the ones which I first encountered on the Texas Instruments microcomputer.

The letter "A" in TI Logo is on tile 65, the letter "B" is on tile 66, etc. Tiles are not available in all versions of Logo. Most versions of Logo do number the letters on the alphabet in the same way, however. The chances are that if you type

```
PRINT CHAR 65
```

the character "A" will appear on the screen. What number would you use to produce the character "D"?

USES FOR THE COMMAND CHAR

The command CHAR is a primitive, or built-in command, in most versions of Logo. Why would you type PRINT CHAR 65? Why not simply type PRINT [A]? The command CHAR can be used to print the alphabet. (In the LCSI version, substitute the word TYPE for the word PRINT1.)

```
TO ALPHABET
   MAKE "NUMBER 65
   REPEAT 26 [ PRINT1 CHAR :NUMBER MAKE
"NUMBER :NUMBER + 1 ]
   END
```

The following Logo procedure allows the characters between any two CHAR numbers to be printed.

```
TO SYMBOLS :START :END
   IF :START > :END PRINT [ ] STOP
   PRINT1 CHAR :START
   SYMBOLS :START + 1 :END
   END
```

In the LCSI variant, TYPE must be substituted for PRINT1, and brackets must enclose the second half of the first line.

```
TO SYMBOLS :START :END
   IF :START > :END [ PRINT [] STOP ]
   TYPE CHAR :START
   SYMBOLS :START + 1 :END
   END
```

To illustrate the use of this program, SYMBOLS 58 62 will print these symbols:

```
: ; < = >
```

The instruction SYMBOLS 65 90 prints the alphabet. The instruction SYMBOLS 97 122 appears to do the same thing. However, if the output is sent to a printer rather than the monitor screen, the characters between

Microworlds continued

65 and 90 print in upper case, but the characters between 97 and 122 are printed in lower case!

```
SYMBOLS 65 90
ABCDEFGHIJKLMNOPQRSTUVWXYZ
SYMBOLS 97 122
abcdefghijklmnopqrstuvwxyz
```

Can you write a procedure that will convert upper case letters to lower case letters when they are sent to the printer?

NON-PRINTING CHARACTERS

Typing PRINT CHAR 33 causes an exclamation mark to appear on the screen. Almost any other number after the command CHAR will also cause a letter or punctuation mark to appear on the screen. However, not all do. Typing

```
PRINT CHAR 7
```

produces a beep instead of a character on the screen. Character 7 is a <u>non-printing character</u>. It produces a noise instead of a letter.

Thirty-two is another number which does not produce a letter of punctuation mark on the screen. It produces a space. This character had been used frequently in past NLX columns in instructions such as

```
PRINT1 CHAR 32
or, TYPE CHAR 32
```

WHICH KEYS HAVE WHICH NUMBERS?

Earlier, the question was posed, "What number is associated with the character D?" Since an A is CHAR 65 and B is CHAR 66, then C is CHAR 67 and D must be CHAR 68. What number is associated with the character L? Counting down the alphabet takes too long, but there is an easier way. Typing

```
PRINT ASCII "L
```

yields the result 76. The word ASCII is a special Logo command which is used to determine the character number of any letter. To verify this, type

```
PRINT ASCII "A
```

and the now familiar 65 appears.

Some of the keys on the keyboard do not have letters or numbers on them. The arrow keys are an example of this type of key. However, these keys do have character numbers associated with them. A one-line program can be used to find out the numbers associated with keys on the keyboard.

```
TO KEY.NUMBER?
   PRINT ASCII READCHAR
   KEY.NUMBER?
   END
```

When a key is pressed, its corresponding CHAR number appears on the screen. This program can be used to determine that a left arrow is CHAR 8, and a right arrow is CHAR 21. What CHAR number is associated with the key labeled ESC?

This knowledge can be used in construction of Instant Turtle programs of the type mentioned in the December NLX. (See "Tipps for Teachers, The Issue of Instant") Instant Turtle programs use single key commands to drive the turtle around the screen. They may take the form:

```
TO INSTANT
   MAKE "COMMAND READCHAR
   IF :COMMAND = "F FORWARD 25
   IF :COMMAND = "B BACK 25
   INSTANT
   END
```

This INSTANT program can be extended to include commands such as IF :COMMAND = "R RIGHT 90. The ASCII command can as easily be employed to use the arrow keys to turn the turtle left (CHAR 8) and right (CHAR 21). (In the LCSI version, instructions such as FORWARD 25 are enclosed in brackets to form a list: IF :COMMAND = "F [FORWARD 25].)

```
TO INSTANT
   MAKE "COMMAND READCHAR
   IF :COMMAND = "F FORWARD 25
   IF :COMMAND = "B BACK 25
   IF ASCII :COMMAND = 8 LEFT 30
   IF ASCII :COMMAND = 21 RIGHT 30
   INSTANT
   END
```

The KEY.NUMBER? program can also be used to identify control characters. For example, if you hold down the key labeled "CONTROL" and press the key "A", the CHAR number 1 appears on the screen. CONTROL-H is CHAR 8, and CONTROL-U is CHAR 21. The left arrow is also CHAR 8, while the right arrow is CHAR 21.

The Apple IIe has a good keyboard which is a pleasure to type on, but the arrow keys are inaccessibly placed on the bottom row beside the space bar. While using the Logo editor, I avoid the problem by pressing a CONTROL-H instead of a left arrow to move the cursor left, and a CONTROL-U instead of a right arrow to move the cursor right.

THE AMERICAN STANDARD CODE

Many languages on both personal computers and mainframe computers use a standard code to store information. In this code, CHAR 7 is a beep, CHAR 8 is a backspace, and CHAR 65 is the letter "A". This code is known as the American Standard Code for Information Interchange.

Since that is a rather long phrase, the term is frequently called by its initials, ASCII. The term "Information Interchange" refers to the exchange of information between two computers, or between two programs in the same computer. For example, a word processor on the Apple which stores its information in the form of ASCII code might be used to write parts of adventure stories which would later be incorporated into Logo programs without retyping them.

The ASCII code is encountered in many places in computing. In AppleSoft BASIC the command

```
PRINT ASC ("A")
```

is the equivalent of the command PRINT ASCII "A in Logo. In the same way, the AppleSoft BASIC command

```
PRINT CHR$ (65)
```

is the equivalent of PRINT CHAR 65 in Logo. These commands are used frequently in BASIC. Recognition of them may make it easier to translate an interesting program from BASIC to Logo.

CODES STANDARD AND NON-STANDARD

Other codes besides the American Standard Code exist. In the past, a code known as Baudot, named after its French inventor, has been used to transmit information between telegraph stations. Baudot Code is still utilized by many deaf individuals who employ teletype equipment to send information to one another over the phone lines.

Another code which is primarily found on IBM mainframe computers, and rarely encountered on personal computers, is EBCDIC. That stands for "Extended Binary Coded Decimal Interchange Code."

Some languages and programs on personal computers do not use American Standard Code. For example, some (but not all) word processors on the Apple use non-ASCII codes in order to make more efficient use of memory. This is an advantage if the program is only to be used by itself, but is a nuisance if it is necessary to share information with other programs. Often, utility programs are provided to covert non-standard information to American Standard Code, but these utilities do not always work perfectly.

CONTROL CHARACTERS

In the American Standard character set, the characters between 32 and 127 are printing characters. Thus, there are 95 printing characters in the standard character set. The characters below 32 are non-printing characters.

If you have a printer, direct the output from the keyboard to it by typing

```
OUTDEV 1 (in Terrapin Logo)
or .PRINTER 1 (in LCSI Logo)
```

Then type the following:

```
PRINT CHAR 12 (in Terrapin Logo)
or TYPE CHAR 12 (in LCSI Logo)
```

There is a fair to good chance that your printer will eject a page from the printer. In business jargon, this is known as a form feed, because it feeds business forms such as checks and invoices through the printer. On some printers the button which moves the paper up one page is labeled "FF".

The non-printing ASCII characters between 0 and 32 are sometimes called <u>control characters</u>, because they are generated by holding down the key labeled CONTROL at the same time a letter is pressed. For example, a CONTROL-H generates a CHAR 8, which results in a backspace on computers which use the standard code for backspacing.

Another reason that these non-printing codes are called control characters is because they are often used to control devices such as printers. That is why CHAR 12 will generate a from feed on most (but not all) printers. Unfortunately, the ASCII code was developed long before the current generation of printers was developed. Thus, printers with features such as compressed mode, or a double-width mode, or a boldface mode were not envisioned. Consequently, each manufacturer has tended to develop its own control codes for its printers.

Fortunately, it is possible to look in the printer manual and identify the control codes for a particular printer. For example, there are more Epson printers than any other brand of printer. The code for compressed print of 132 characters per line (rather than the standard 80) on the Epson MX-80 printer is CHAR 15. The code for turning off the compressed print is CHAR 18. If you have an Epson MX-80 printer, try sending the following program to the printer.

```
     TO DEMO
        PRINT CHAR 15
        PRINT [ THIS IS IN THE COMPRESSED
MODE.]
        PRINT CHAR 18
        PRINT [THIS IS NORMAL PRINT.]
        END
```

If you have another brand, examine the manual to identify the control codes which operate the features of your printer. There is a hardware company which manufactures a row of buttons which can be attached to a printer to control the various printing features. As I recall, the retail price of the device is about $60. You should be able to achieve the same result with a two-line Logo program, and save $60.

PRINTING REPORTS

In last month's NLX, I described a method for printing the text of a procedure. I developed the method for printing Logo adventure stories, but suggested it might also be used to print letters to parents and end-of-school-year reports. In next month's NLX, I will examine how control characters in these reports can be used to underline text, high-light figures, and enhance the text in other ways.

Glen Bull is a professor in the University of Virginia's School of Education, and teaches Logo courses at both the graduate and undergraduate level.

Right Here In Turtle City

by Steve Tipps

"Trouble in Logoland" is the title of Mary Cron's article in the December 1983 issues of TLC magazine. The title is misleading in that the article is a well reasoned discussion of the results from the Bank Street and University of Haifa studies.

The results of these studies appear to refute claims about Logo made by Papert. Complex ideas in Logo did not transfer into other situations or to other types of problems, according to the researchers. In addition, knowledge about Logo programming was not as high as the researchers expected.

According to Cron, the results of these studies were predictable from the instructional methods used. The researchers set up non-directive learning situations in which teacher intervention was minimal. She feels that this method is not appropriate for Logo, preferring instead, Guided Discovery. In this method, the teacher is a resource who has both directive and facilitative roles.

LOGO GUIDANCE

My Logo experience and that of others (see "The Hazards of Hacking" by Sharon Burrowes, January 1984 NLX) coincides with Cron's. The students who make the most progress with Logo have teachers who are willing to step in to prevent discouragement and to uphold standards.

Students do not spontaneously generate procedural problem solving, recursion, variables, or list processing. What they do generate are situations in which these concepts would be useful. A developmental approach grows with their needs rather than leaving them to their own devices. Students do not always know when to ask for help.

Guided Discovery as the needed approach will not come as news to any one who has been involved in other active learning programs: kindergarten, mathematics and science laboratories, Elementary Science Study, Science Curriculum Improvement Study, Minnemast, Outdoor Biological Instructional Strategies, Language Experience Approach to Reading, Independent Reading Programs, Gifted and Talented Programs (Treffinger or Renzulli models), Moffett's Interact Language program, Graves' writing programs, ad infinitum. Guided Discovery has been the recommended instructional strategy for the last twenty years, but still has not become the prevalent teaching method.

There is a good reason for our practice to fail our preaching. Guided Discovery teaching is the most difficult of all teaching strategies. It requires teachers who are well grounded in the "content" being covered, and who understand that instructional objectives may be attained in many ways. It takes teachers who are self assured about when to remain quiet and when to intervene, and who are willing to allow themselves and students to make mistakes in the pursuit of knowledge.

These teachers have two ultimate goals for students: success and independence. The method is linked directly to the goals.

LOGO TRANSFER

Mary Cron is not surprised that non-directed teaching resulted in no overwhelming knowledge about Logo or transfer of problem solving skills. Laissez faire, non-directed teaching mimics guided discovery teaching, but it is not the same at all. The laissez faire teacher has little direction and gives little direction.

Many people wrongly believed that totally non-directed teaching was the basis of the open classroom. However, good informal classrooms were always structured learning environments with provision for individual learning needs.

The demise of the movement towards open classrooms occurred not because they did not work, but because they were too much work. Only places which were willing to provide constant leadership and training have sustained informal learning situations. The informal classes in Upper Arlington, Ohio, and Drew Model School in Alexandria, Virginia, are examples.

In the world of educational research, generalization of learning from Logo to other situations would have been miraculous rather than inevitable. Transfer of learning is a rare finding. Even had the methodology been closer to Guided Discovery than non-directed, I wonder if the results would have been different.

Perhaps the claims for Logo have been too high and too idealistic, as Cron suggests. Or, perhaps the way we generally evaluate learning with group norms and standard deviations is not appropriate for the kind of highly individual learning which goes on with Logo.

LOGO TESTS

The low scores on a paper and pencil test at Bank Street were similar to the results of a paper and pencil test we tried out last year at the University of Virginia. However, even students who did poorly on the quiz had a wide range of skills on the computer.

Richard Noss, director of the Chiltern Logo Project in Hatfield, England, expressed amazement at the way educational research is conducted in the United States. Giving a paper and pencil test on Logo skills seems to him, and to me, quite similar to giving a paper and pencil test on riding a bicycle. The Bank Street conclusion was that children did not learn much Logo; our conclusion was that the evaluation technique we used was inappropriate.

In addition, many of the ideas beyond turtling in Logo depend heavily on abstract reasoning. Although children can manipulate ideas such as variables and recursion at a concrete level (see Joan Rainey's article elsewhere in this issue), they understand them only in terms of tangible effects. Many problems depend on abstract manipulation of ideas such as variables, combinations, permutations, correlations. These relational ideas do not develop quickly.

Seymour Papert suggests that work with Logo will speed this intellectual development. I don't believe that two years of Logo will overcome eons of human development. I do believe that Logo provides a new environment in which learners (of all ages) can play and manipulate ideas in a way never available to them before.

Turtle City continued

As a result of this play, the advent of abstract reasoning may be considerably eased. If Logo has impact on these formal thinking abilities, the differences will be measured in days and weeks, not in years.

SHOW ME

A frequent query I receive is, "Show me studies which prove Logo teaches problem solving, mathematics, language arts, etc." I reply, "Show me studies which prove math books teach mathematics, pencils teach social studies, overhead projectors teach reading." The standards to which Logo is held seem different from those expected of other instructional materials.

People prefer a quick fix to problems. But neither Logo nor computers in general are likely to solve educational problems just through their existence. They are tools which may help in solutions.

Tools require competent users. Computers do not replace teachers or diminish their importance. More highly qualified teachers are needed to structure learning with the computer as another instructional tool. Dave Morsund's editorial in The Computing Teacher's special issue on Logo says it well:

> "Attracting and holding good teachers, providing them with high quality preservice and inservice education, supporting them with appropriate resources--these are keys to improving our educational system. I believe in the potential of Logo, but I believe much more strongly in the part educators will play in its effective use."

WHAT TO DO?

The effects of Logo should be examined in more detail. Studies of limited scope, with appropriate guided discovery teaching, are needed. The aim would not be to defend Logo, but to determine how Logo can best be used. Only in such situations can we find out what Logo does for learners.

Mixed research designs with anecdotal case study material need to be balanced with more objective measures. But, failure to winnow robust statistical results does not invalidate the goals or methods of Logo.

We must be careful when listening to the likes of Professor Harold Hills (Music Man) proclaiming that we have "trouble in Turtle City." Our trouble is that we don't have enough good programs and experience to know how Logo works best.

Papert raises our eyes to the stars, but does not provide many hints about how to get there. The task of working with Logo is left to mortals who have much to learn.

Steve Tipps is a regular columnist for The National Logo Exchange, and frequently contributes additional articles.

Logo In The News

inCider magazine for January 1984 features an article by Molly Watt which chronicles the adventures of Joey's drawing with Logo. In "Logo Building Blocks," she tells how Logo responds to Joey's needs and questions as he makes more elaborate designs. Teachers will find encouragement in the way Logo (and the sensitive Logo teacher) allow children to find their own directions in the Logo environment.

This article is one of a series of four related to Logo which will appear in inCider in the coming months. February's article will feature debugging, March's will make a comment on "no threshold, no ceiling," and the one in April will tell about computer cultures. Be watching for them.

Molly would appreciate any comments from NLX readers. Send them to her at Educational Alternatives, Gregg Lake Road, Antrim, NH 03440.

A+ magazine (December, 1983) has an article called "Follow the Turtle to Logo" by Steve Rosenthal. The article reviews most of the fundamental commands of turtle graphics, discusses some of the structural issues of Logo, but makes only a short mention of words and lists. The sidebars on the article mention some of the antecedents to Logo and the Apple Toolkit.

Logo News

Eric Sandberg-Diment writes about "Logo, The Friendliest Language," in November 1983 Science Digest. A quotable quote: "What makes Logo such a good first language is its forthrightness. The language doesn't get in the way of what you are trying to do."

"The Very Thought of Turtles" is reprinted in Readers Digest January, 1984 from Yankee magazine. Although not about Logo, this personal story of box turtles and their behavior might lead to your Logo students becoming more interested in studying "real turtles." Sounds like a dynamite name for a television series!

A 100 page Logo magazine arrived at NLX from Japan. The publication is written by an active Logo user's group in Japan and is called Logo Some Z's. The issue is entirely in Japanese except for an occasional English word and Logo commands.

The magazine looks fascinating, and includes an article on playing a flute and reviews of several different Logo versions (Epson, and others), books (Learning with Logo and Turtle Sourcebook), and other Logo materials.

A chart on the inside cover features a genealogical tree of Logo, with its roots in artificial intelligence. According to the outermost branches, there are many versions of Logo being developed in Japan.

For more information, contact Setsuko Abe, Geodesic Inc., 401 Park Heights, 1091 Kumagawa, Fussa-shi, Tokyo, Japan.

Edward H. Dougherty invites all Logo students and teachers to submit their favorite procedures for publication in a new book he is writing about learning Logo. He is especially interested in projects which include comments and stories by students telling about the development of the procedures. For more information, write to Edward at 1501 Presidential Way, Suite #16, West Palm Beach, FL 33401, or call (305) 684-4770. You may also leave an email message on CompuServe for 74216,3236.

A report on True BASIC being developed by John Kemeny and Thomas Kurtz (the originators of BASIC) in Infoworld (January 16, 1984) claims that it will possess "graphics capabilities which surpass those of Logo." What about those words and lists and recursion and ...

The move to develop software with Logo or to be Logo-like goes on. Carousel Software announces three new programs - Telly Turtle, Brain Strainers, and Simulated Computers - which they say are based on the Logo language. (Infoworld, December 26/January 2, 1984, p. 43.)

COMPU-LIT (PO Box 254, Sherborn, MA 01770) has already released Logo Locomotion and Ticktock which were developed with Logo. An unpublished review of these two programs by Tom Muncy of Waynesboro, VA, points to some potential problems. Ticktock was extremely slow in operation (50 - 90 seconds per clock face). Use of PICTs would have speeded this up considerably. The skills presented in Logo Logomotion rely on coordinate geometry and are not as "intuitive" as the more typical Logo sequence. The reading level of the instructions was fairly high (7th grade, Fry) for a program which would be intended for introduction of turtle graphics.

Development of instructional design for CAI with Logo is certainly possible. Implementation of commercial CAI with Logo may not be as feasible.

Dave Boyd, of Synetix Inc., wants to inform all NLX subscribers that the Terrapin sprite drivers now available from Terrapin Inc., 380 Green Street, Cambridge, MA 02139, for $30, will support their series of sprite boards, if they are plugged into Apple slot # 4 instead of # 7, as suggested in their documentation. Prices for the sprite boards range from $149 to $395. For more information, write Synetix Inc., 10635 NE 38th Place, Kirkland, WA 98033, or call (800) 426-7412.

NLXual Challenges

with Robs Muir

This month's problem is a special challenge. First, let us create a problem which uses a circle of some known circumference. For example, we can draw a circle with the familiar statement REPEAT 360 [FD 1 RT 1].

If you remember your geometry, you may (?) recall that the circumference is equal to the diameter times pi, or C = D * pi. This formula can be rewritten to include the radius, rather than the diameter. Since the radius of any circle is twice the diameter, our formula becomes C = 2 * R * pi. All we need now is the value for pi and we are ready to proceed.

If you attempted the problem of last month (the circle in a box), you probably encountered the value of pi. Most of the solutions submitted used some variation of pi, the number 3.14159.

Indeed, I received an excited late night call from a colleague who had experientially derived pi in the course of solving this problem. He kept coming up with a value approximating 3.14159, when, suddenly, he made the connection! My skeptical friend, who holds a PhD in education, finally was convinced that Papert's vision of discovery based learning in computational environments works. Another convert!

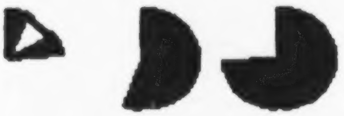
Armed with an understanding of the relationships among the circumference, diameter, and radius of a circle, you should be ready to tackle the next problem. Draw our circle of REPEAT 360 [FD 1 RT 1] and fill it in.

There are many possible ways to accomplish this; some are efficient, some not so. Perhaps we should define efficiency in terms of speed. Here is a very inefficient method.

```
TO FILCIRC :SIZE
   REPEAT 360 [TRI :SIZE RT 1]
   END


TO TRI :SIZE
   REPEAT 3 [FD :SIZE RT 120]
   END
```

Can you write a FILCIRC procedure that will fill in a circle of any size? How fast will your method fill a circle with a diameter of 114.592 (360/3.14159) turtle steps?



During my attempts at a satisfying (and efficient) solution, I used several different approaches. With Terrapin Logo, my fastest method filled in our "standard" circle in 40.2 seconds. The identical method in Apple (LCSI) Logo required 58.2 seconds! Can you fill in the NLX circle faster?

Note: As with all NLXual Challenges, if you want to receive a possible solution, send a copy of your solution (or your attempt) with a <u>stamped</u> self addressed envelope to NLXual Challenges, Attn: Robs Muir, 1688 Denver Avenue, Claremont, CA 91711. We will compile contributions and send you the most elegant solution.

If you have an idea which might be developed into a challenge for this column, send it to me at the above address. Thanks!

Robs Muir is a teacher in the Claremont CA Unified School District and Claremont Graduate School. He presents Logo workshops in the southern California area.

The National Logo Exchange, copyright 1984 by Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly, September through May, $25 (US) per year, mailed FIRST CLASS from Charlottesville, Va. $5 (US) additional per year for addresses outside the USA, Canada, and Mexico. The opinions expressed by the authors are not necessarily those of The National Logo Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress Street, Salem, MA 01970. Address other correspondence to: The National Logo Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor..Tom Lough

ISSN 0734-1717

The National LOGO Exchange

Volume 2 Number 7 FORWARD 100! March 1984

The Turtle Goes to College

by

Lois Tepper

Last summer, I offered a Logo course at Southampton College, New York, which was taken by eight Masters of Education level teachers. With one exception, the teachers "knew nothing about computers," but all were extremely eager to learn. Yet, some were skeptical that the computer would be little more than an expensive ditto machine. But then, that was before they knew about Logo.

The course was designed to begin with a discovery based approach to turtle movements, then move to programming simple geometrical shapes, discussion and implementation of top down programming to produce more complex designs, and, finally, into a discussion of list capabilities.

GETTING STARTED

Admittedly, my teachers started out with some degree of trepidation. To help overcome this, I had them team up in pairs.

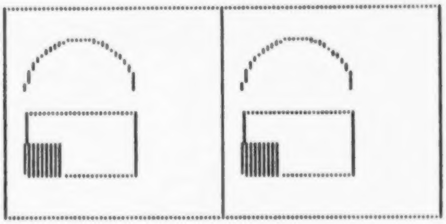
By the second session, everyone was drawing elementary geometrical shapes and experimenting wholeheartedly with the turtle.

Already, I noticed that the teachers who knew how to type were experiencing a degree of frustration because of the special computer keys, and the differences in the keyboards of the Apple II and the IIe. As they progressed, however, their keyboard skills increased, and the frustration began to be replaced with "Ooh!" and "Aah!"

TEACHER DEMANDS

Soon, my teachers became more demanding. Several wanted to use Logo to write tests and CAI programs. I felt I should use this motivation as soon as possible. Others wanted to work more on their graphics projects.

All demanded more time on the computers! I had to arrange for computer access eight hours per day, with two additional three hour blocks before class, plus two more hours after class! Their enthusiasm for their Logo work was fantastic!

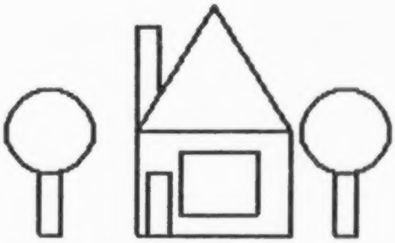


As they progressed, I became aware of peaks and valleys in their learning. Often, I would hear the "Ah-ha!" of a peak experience, when something clicked. It was fun to observe the insight gained when they learned the concept of writing individual procedures for separate tasks and using a superprocedure to call them.

The valleys of frustration were quickly bridged with ready cooperation and support from teammates and other class members.

IS IT OVER ALREADY?

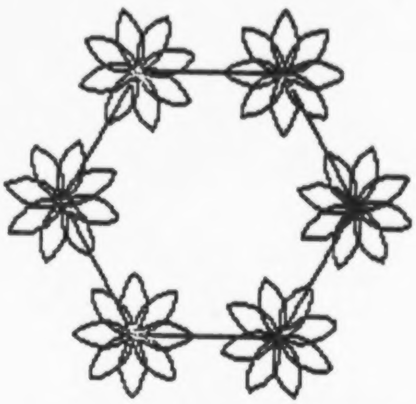
The end of the class came all too soon. As I took stock of what we had done, I became aware that a very special thing had happened.



College Turtle continued

Certainly, the teachers has been introduced to computers in a meaningful way with Logo. They had created and solved problems in which they had personal interests, and experienced how Logo could introduce their own students to problem solving.

But the special thing was how the class of eight strangers had become eight friends, exchanging phone numbers, arranging camping trips, causing one class member to observe, "I have never been in a class where the members grew so close to each other!"



Logo is like that. Their beautiful social experience underscored the importance of cooperation between children in the Logo problem solving process in a way no lecture or handout could ever do.

But the thing I personally treasure the most was one of the greatest things that can happen to a teacher: they all said over and over how they still wanted more!

Lois Tepper is an Assistant Professor in Experimental Psychology at Southampton College, Long Island, NY. She is currently studying applications of Logo to special populations.

Here's Adam

Coleco's Adam Computer has arrived. According to a recent article in Infoworld, SmartLogo, developed by Logo Computer Systems Inc. (LCSI), will be available. No date was given. Rumors have it, however, that beta site testing has already begun.

Conference Information

The Arizona State University College of Education announces its fourth annual conference, "Microcomputers in Education," to be held March 15-16. For more information, write Dr. Gary Bitter, College of Education, Arizona State University, Tempe AZ 85287, or call (602) 965-7363.

The professional journal, Computers, Reading and Language Arts, is sponsoring an eastern states conference entitled "Computers & Reading / Learning Difficulties," April 7-8, in Philadelphia, PA. Joyce Tobias, Curriculum Coordinator of Computer Education for Brookline (MA) Public Schools, will deliver a presentation, "Logo and the Language Arts," which will include implications for the teaching of reading and writing with Logo. Krell Software Corporation will give a workshop on improving conceptual thinking with Logo. For more information, call (415) 222-1249, or write Educational Computer Conferences, 1070 Crows Nest Way, Richmond CA 94803.

A national conference entitled, "Technology for Disabled Persons," will be held in Chicago, October 1 - 3. The deadline for proposals is June 1, 1984. Conference presenters will receive a 50% discount off the registration fee. For more information, write to Office of Continuing Education, University of Wisconsin - Stout, Menomonie WI 54751.

A Contest for Logo

The University of Wisconsin - Parkside announces its fourth annual International Computer Problem Solving Contest, to be administered on Saturday, April 28. Problems involving computation, simulation, patterns, words, and a "mind bender," are offered to three divisions: elementary (grades 4-6, junior (grades 7-9), and senior (grades 10-12). Any computer language may be used in the contest. This is an outstanding opportunity to show what Logo can do. To receive more information on how your school can become a local contest site, write to D. T. Piele, ICPSC, Box 2000, University of Wisconsin - Parkside, Kenosha WI, 53141. A set of problems and solutions from the 1983 contest is reported in The Computing Teacher, October 1983, pages 21 - 26.

From The Editor

If you have followed college basketball the last few years, you know the name of Ralph Sampson. Standing 7 feet 4 inches tall, he recently turned professional, after a brilliant four years playing for the University of Virginia.

A joke which has made the rounds of Charlottesville goes like this:

A. Knock! Knock!
B. Who's there?
A. Ralph.
B. Ralph who?
A. How quickly we forget.

Yes, sometimes we forget rather quickly the way things were. Why, we even forget the way WE were!

Seymour Papert tells of an experiment he performed once, in which he filmed a group of children every three months as they worked a set of Piagetian tests on conservation.

For example, they were shown a glass of water, and watched it being poured into a slimmer taller glass. As expected, the four year old children reported that there was more water in the taller glass.

When the children got old enough to "know" that the amount of water was the same in each glass, Papert asked them if they remembered their earlier responses. The children indicated that they had always said the amount of water was the same.

When shown films of themselves saying the amount of water was different, the children were unable to accept that they had ever said anything except that the amount of water was the same. In fact, they thought the films were fake!

How quickly we forget, indeed!

As adults, let us strive to NOT forget how we were before Logo. Let us not forget our enthusiasm, our confusion, our hopes, our frustration, our accomplishments, our discards.

We are in position to help many other adults along the Logo path. The colleagues who are learning Logo now may not be as confident, as enthusiastic, as hopeful as we believe we were. Our ability to remember our own initial encounters will make better teachers of us all.

FD 100!

Tom

NLXionary

A lectionary of selected Logo readings

compiled by Griff Wigley

Welcome to the NLXionary! Each month, we will present a lectionary of articles on Logo from other publications for your reading pleasure.

RECENT

Family Computing Magazine, February 1984, "Learning Logo is a Family Affair," by Mindy Pantiel and Becky Peterson, "Different Versions of Logo," by Kenneth P. Goldberg, and "The Man Behind Logo," by Nick Sullivan.

>These three articles form a neat, concise overview of Logo, directed mainly at the home user. If I were to present Logo for the first time to a group of parents, I would steer them to these articles. For the Logo shopper, the Logo comparison article and chart are among the most up to date and accurate I've seen. Logo teachers and Seymour Papert fans will find Sullivan's article and interview the most interesting of the three pieces. New bits of Papert's current thinking are scattered among familiar philosophizing.

Classroom Computer Learning, October 1983, "Logo in Special Education," by Susan Jo Russell.

>This is a jewel among Logo articles. The author clearly is both a thinker and a teacher. Her introduction on computers as timesavers poses an interesting new challenge to the typical use of software in special education classrooms. Two scenarios with handicapped children are used to further illustrate her contention of Logo's usefulness for finding alternative methods of learning and teaching.

NLXionary continued

Monitor, July / August 1983, "Children, Computers, and Logo," by Dr. George P. Young.

>Here is a Logo article written by the superintendent of the St. Paul, Minnesota, Public Schools. It is just what you need to lay on the desk of your friendly administrator (accompanied by your requisition for Logo related goodies, of course!). The article describes the extensive Logo project in St. Paul, followed by teacher testimonials. The last paragraph contains his observations on how Logo using teachers are changing their whole way of looking at teaching and learning.

ANCIENT GEMS

Popular Computing, September 1982, "Should Children Be Computer Programmers?" by Dan Watt.

>In this "relic" from the past, Watt answers his own question with a clear YES. Read his comments carefully, however. Problem solving, creative power, and a deeper learning of content are the benefits; job preparation is not, for the most part. Although Logo is not the main topic, this might be just the article you have been seeking for developing goals and objectives in your curriculum with Logo.

MENTIONABLES

Family Computing, March 1984, "Rendezvous with a Robot."
T.H.E. Journal, February 1984, "Applications" column.

>Both articles explore the use of Logo and other languages with Androbot Company's early version of TOPO the Robot.

Compute!, January 1984, "The Demons of Atari Logo," by David D. Thornburg.

>This article is an exploration of the WHEN demons, a feature of Atari Logo used in collision detection.

Griff Wigley is co-owner of Family Computing, Inc., a facilitator in the Faribault (MN) Public Schools, and a school board member of Prairie Creek Community School.

Papert Addresses Boston Logo Users' Group

reported by
Nola Sheffer

There was an air of anticipation and excitement at the Boston Logo Users' Group in November, for Professor Seymour Papert was to be the guest speaker. The meeting site had already been changed to accommodate the crowd of over 200.

For his topic, Professor Papert chose to address some of the criticisms of Logo. He prefaced his talk by saying that many of the complaints were not sensible, or reflected an ignorance of the language. Therefore, he would choose to address just two classes of criticism: that of the philosophy, from the perspective of cognitive psychology, and that of the programming perspective. Is Logo the best language to use beyond the turtle graphics? Are there other languges which provide an easier entry into words and lists?

COGNITIVE CRITICISM

An example of the first type of criticism comes from Roy Pea, a cognitive psychologist connected with the Bank Street College of Education. He decided to test the claim that Logo can be learned without teaching. He put children in a classroom with Logo and computers, and observed the results. In attempting to measure knowledge gained as a result of working with Logo, Pea assumed that using Logo would result in increased planning ability, and tried to measure that.

The result: No apparent increase in planning ability was shown by students after using Logo.

Quite apart from any concerns about the means of measuring planning ability, Papert questions the assumption that learning Logo results in increased planning skills. In fact, Papert asserts that, if there were a true Logo environment, scores on "planning" might actually decrease! Those students who were planners to begin with, would continue to be planners. But, those who were weak in this area might decline because Logo would be the one area in school where a "top down" approach is not pushed, and, thus, the non-planners might feel "liberated" and actually do less planning than before.

Papert continued

What learning to program will do for an individual will depend upon that individual. Not everyone needs to gain the same benefits. A non-planner might benefit the most from using Logc because the environment would allow him/her to work in a style that is comfortable - an option not generally available in the typical school.

In the case of programming, Papert believes that much of the criticism of the non-graphics part of Logo can be answered by finding better ways to connect graphics and non-graphics. One way of bridging the two is through sprites.

THE SPRITE CONNECTION

To talk to a particular group of sprites, it is necessary to put them in a list and name them. This gives a nice introduction to both variables and lists. A command can now be given to the whole list, or different commands can be given to each sprite in the list, using recursion to work your way through the list.

Also, Professor Papert thinks that recursion is not so confusing if you have already presented a good model for the calling of subprocedures. One of the models he likes to use is that of "actors," that is, having different people "act out" the work of particular procedures.

HOW MUCH TEACHING?

A question about the amount of teaching that is needed for Logo was raised. Papert responded that very little teaching might be needed at a place such as the Lamplighter School in Texas, where programming is part of the culture, and the children are used to an informal atmosphere.

Papert saw it as an advantage that, during the first year of using Logo at the Lamplighter School, there were not many teachers available with a knowledge of the language, thus forcing students to use peers, as well as their own resources.

This is NOT to say that it is not necessary to have knowledgeable people around; students do need access to expertise at appropriate times. The issues of when to intervene, and how much intervention is needed, are two of many that Professor Papert will be examining in a project in the Boston Schools.

Apart from an abundance of computers in the classrooms, this Boston project will be an environment quite different from that of the Lamplighter School in suburban Dallas. Unlike the Texas students, these will be students from typical inner city public schools, as opposed to a private school with an informal atmosphere.

It will be very interesting to see what differences, if any, there are either in the learning or the teaching of Logo!

Nola Sheffer is Co-Director of the Logo Users' Group of the Boston Computer Society.

TIPPS for TEACHERS

by

Steve Tipps

Triangle Thinking :SIDE3

Searching for an all purpose triangle has been a goal of the last two columns on triangular thinking. In this pursuit, other triangular ideas have been encountered. This xmonth, a possible solution to the problem posed in the February column will be offered. The goal of this problem was to find a relationship between the sides of various triangles and to make a transportable right triangle.

The Type III problem in last month's column would probably be appropriate for seventh graders through college mathematics students. The point of the inquiry would be the discovery of the trigonometric sine function.

I remember learning to use the sine and cosine tables in trigonometry and geometry, but I do not remember ever knowing where they came from or that I had any ability to understand them. For all I knew, the table was of mysterious origin, probably set down by the Greeks or Egyptians from divine inspiration.

A TYPE III problem provides a context for exploring and deriving meanings in math or science or language arts. (See Bull /Tipps, Problem Spaces in a Project-Oriented Logo Environment, NLX Jan-Feb 1984 and The Computing Teacher, Dec 1983-Jan 1984 for descriptions of other problem types.)

Tipps continued

AUTO.TRI REVISITED

In the previous columns, a procedure named AUTO.TRI made right triangles of any size with HOME as an anchor. The Pythagorean theorem was used to calculate the length of side 3 (the hypotenuse). A suggestion was made to find the heading of the turtle at each angle. From this information, the possibility of a relationship between the length of the sides and the angles was suggested.

Working through this problem involves many stages. The first step involves finding a relationship between the angles and the sides.

Holding some of the variables constant is essential; making the inputs to AUTO.TRI equal is a good way to simplify the possibilities. Then addition, subtraction, multiplication, and division of the sides may be attempted. Arithmetic operations with variables can be attempted to find a pattern of results.

```
:SIDE1 + :SIDE2
:SIDE3 - :SIDE1
:SIDE1 * :SIDE3
:SIDE1 / :SIDE3
```

Finally, a pattern involving division may be discerned so that, when the sides are equal, the angles are always 45 degrees, and the quotient of :SIDE1/ :SIDE3 and :SIDE2/ :SIDE3 is about .707. This is an important hook into the entire problem.

The information from the different AUTO.TRIangles can be recorded, or a procedure for printing the information can be written. For my solution, I wrote a procedure which I called PRINT.INFO to organize the information about the sides, the angles, and the relationships. Likewise, I wrote CHART.HEADING to set up the form.

Getting the sides, angles, and divisions printed out is a good project in itself. The angles are actually calculated to three or four places and the quotients of sides are to seven or eight decimals places. ROUND is a primitive which rounds off a decimal number to the nearest whole number. For example, PRINT ROUND 4.3637732 will return the number 4.

ROUND can be used in a procedure to truncate a decimal fraction to any number of places. For example, if you wanted to round the decimal number 0.65378 to two places, you might first multiply it by one hundred, ROUND the result, then divide it by one hundred. I chose four places for my chart, and called my procedure FOUR.PLACE.

SIDE1	SIDE2	TURN SIDE1/SIDE3	SIDE3	TURN2 SIDE2/SIDE3
30	30	45 .7071	42	45 .7071
50	50	45 .7071	71	45 .7071
90	90	45 .7071	127	45 .7071
30	60	27 .4472	67	63 .8944
60	30	63 .8944	67	27 .4472
45	30	56 .832	54	34 .5547

What has been generated is a sine table for the angles. The accuracy of the table can be compared with another function in Logo. Compare the quotients of the sides with the results using the SIN primitive.

```
PRINT SIN 45
PRINT SIN 63
PRINT SIN 34
```

Also compare the values of the sine functions of Logo to a sine table.

Other ideas which might also be developed from the chart are that the sum of the two angles is always 90 degrees and the quotients are always less than one. An extension would be to graph the values of the sine for all the angles from 0 to 90 degrees. This could be done on paper or with a program written to plot the points with Logo.

MOVING TRIANGLES

A good TYPE III problem not only has content embedded in it. It also allows you to do something meaningful. The ultimate goal of this exploration was to have a movable right triangle anywhere on the screen. When

more Tipps

the sine values are calculated from the sides, the angles of the triangle are set by turning toward HOME. To have a movable triangle, the angles must be figured from the sides without being dependent on HOME. The challenge is to find the angle associated with a certain sine quotient.

One strategy to approximate the angle would be to turn the turtle slowly from a heading of 0. When the sine of the new heading is greater than the value found by dividing :SIDE1 by :SIDE3, the heading is equal to the value of the interior angle needed to close the triangle with :SIDE3. This procedure tests the sine of the turtle heading against the needed sine.

```
TO ANGLE? :SIDE1 :SIDE2
    HT
    TEST  SIN HEADING > FOUR.PLACE :SIDE1  /
:SIDE3
    IFTRUE [OUTPUT HEADING STOP]
    RT 1
    OUTPUT ANGLE? :SIDE1 :SIDE2
    END
```

This procedure is one crucial step in getting the turtle to determine the angle which it needs to finish a right triangle. The turtle also has to remember its heading before it makes this judgement and adjust that heading by the amount of the angle discovered. When that maneuver is completed, all the missing pieces are in place for an all purpose, any place, any time right triangle.

I'll leave that as your next puzzle. If you would like a copy of my solution, send a self addressed stamped envelope to me at the NLX, PO BOX 5341, Charlottesville, VA 22905, and ask for the Movable Triangle Procedures. While you're at it, why not enclose your solutions as well? I'd like to see them. Thanks!

Steve Tipps is a professor in the University of Virginia's School of Education, and presents Logo workshops for school systems throughout the United States.

Allen's Book is Something to Think About

A review by

Glen Bull

Four years ago, a book appeared which changed my career. Seymour Papert's Mindstorms: Children, Computing, and Powerful Ideas contained thought-provoking ideas about education. Its treatment of Logo invited me to investigate, and I was captured. I have given away over 20 copies to colleagues. I even placed a copy in the hands of Steve Wozniak (designer of the Apple computer) as he got on the plane to leave after speaking at the University of Virginia's School of Education.

Two years later, a language manual, Apple Logo, by Hal Abelson, was published. This book explified what a language manual should be. It was spiral bound, so that it would lie flat while I was typing at the computer. It had a clean layout, and illustrations at the right points to help me follow the text. Best of all, Abelson's explanations of Logo's commands were superb. Apple Logo promptly went on my bookshelf beside Mindstorms.

Now I have placed a third book beside the first two. I first encountered Thinking about TLC Logo (John R. Allen, Ruth E. Davis, and John F. Johnson, New York: Holt, Rinehart, and Winston, 1984) last spring at the National Educational Computing Conference in Baltimore. I made a considerable nuisance of myself on the way back to Virginia with Steve Tipps and Tom Lough, because I would not stop reading sections of it out loud. Threats of physical violence (which I abhor) were finally used to quiet me.

Thinking about TLC Logo, or TATL (pronounced "tattle") as it is sometimes called, is an irreverent, freewheeling discussion of sophisticated computational ideas. So that you will know my bias, I will report that Thinking about TLC Logo appeals to me at three levels.

Allen's Book continued

BOTTOMLEVEL

The first level is that of the lowest, most contemptible form of humor, that of word play. One of the turtles in TLC Logo has the power to make copies, or clones, of himself. Naturally, this turtle is known as the clone arranger, with a suitable picture of the masked marvel in the text. TATL is filled with such double and triple meanings.

MIDDLELEVEL

Graphics are the second aspect of the book which captured me. I don't know if there is a term for graphic puns, but there are not only turtles, but ducks, rabbits, and humans (the authors?) in illustrations of visual humor throughout the book. The visual element contributes to a general feeling of irreverence.

The first edition I saw was a special version for the Aquarius computer. One of author's names was deliberately printed upside down on the cover. I regret to say that this has been rectified in a retreat to reality in the more general edition.

TOPLEVEL

The third element of the book caused me to leave it on my bedside table for some time. Thinking about TLC Logo includes discussions of elegant ideas. Ideas are the heart of this book. The subtitle of Mindstorms is "Children, Computing, and Powerful Ideas". The subtitle of TATL is "A Graphic Look at Computing with Ideas". Both books are dedicated to the concept that thinking and learning can be fun, and both are centered around the use of Logo as a tool to explore those ideas.

The TLC in TLC Logo stands for "The Lisp Company," the name of Allen's software development business. However, TATL was written in a "generic" style, so that it can be used with any Logo version.

TATL is influenced by historical perspectives from artificial intelligence and LISP. (LISP is a computing language used for artificial intelligence studies, and is the direct ancestor of Logo.) TATL devotes considerable space to questions of deducktive logic. It brings with it a heavy infusion of ideas from LISP which properly belong to Logo as well.

I would not advocate placing a copy of Thinking about TLC Logo in every elementary classroom. The book is not written at a third grade level. I do believe that every one who is holding a workshop for teachers on Logo should have a copy of this book, so that they obtain a feeling for the proper breadth and perspective of Logo.

Glen Bull is a regular columnist for The National Logo Exchange, and frequently contributes additional articles.

Logo In The News

Molly Watt's education column in the February issue of inCider magazine is entitled, "De-Bug Collection." If you have been looking for ways to help your students use their "mistakes" to good advantage, don't miss this informative article. Molly traces the struggle of Tanya and Bill as they attempt to draw a boat with Logo. She encourages adults to become comfortable with the everpresent Logo programming bugs, and share them with their students. Reading this piece brings back memories of the lessons suggested by Nancy Sopp in her near-classic "Logo Mess Ups: Process vs. Product," from the NLX, October 1983.

On page 78 of February's Instructor magazine, Joyce Tobias mentions her Logo classes in the Brookline (MA) Public Schools which are run on a network of computers with a hard disk system.

A review of DR Logo appears in the February 13, 1984 issue of Infoworld magazine. All categories examined were rated excellent. Also in the same issue, Tom Snyder, president of an educational software development company, suggests that, in an ideal world where the effects of Logo could be properly tested, we would find that the teachers changed more than the students!

An article in ISO World, November 14, 1983, page 105, outlines the plans of several companies for taking Logo out of the classroom and into the office. "Logo Isn't Just for Kids Anymore" documents support from Digital Equipment Corporation (DEC) and Digital Research Inc. (DRI) as evidence for more Logo utilization in business applications.

MICROWORLDS

by
Glen Bull

Centering Titles

In the January 1984 NLX, I described a method for using the Logo editor to write text to be printed on the screen. This month, I will continue our list operations exploration with the development of a procedure to center titles. This procedure can be used with the WRITE procedure of January, reproduced below.

Terrapin Version
Apple or Commodore

```
TO WRITE :PROCEDURE
   WRITE.LINE BUTFIRST TEXT :PROCEDURE
   END

TO WRITE.LINE :PROCEDURE
   IF :PROCEDURE = [] STOP
   TEST LIST? FIRST FIRST :PROCEDURE
   IFTRUE RUN FIRST FIRST :PROCEDURE
   IFFALSE PRINT FIRST :PROCEDURE
   WRITE.LINE BUTFIRST :PROCEDURE
   END
```

LCSI Version
Apple or IBM

```
TO WRITE :PROCEDURE
   WRITE.LINE BUTFIRST TEXT :PROCEDURE
   END

TO WRITE.LINE :PROCEDURE
   IF EMPTYP :PROCEDURE [STOP]
   TEST LISTP FIRST FIRST :PROCEDURE
   IFTRUE [RUN FIRST FIRST :PROCEDURE]
   IFFALSE [PRINT FIRST :PROCEDURE]
   WRITE.LINE BUTFIRST :PROCEDURE
   END
```

These procedures allowed a block of text to be printed without resorting to a series of PRINT statements. They also allowed commands enclosed in brackets to be placed on a line, as in an adventure story setup.

```
TO INSTRUCTIONS
  [TAB 6]
  ADVENTURE STORY INSTRUCTIONS
  [BLANK.LINE]
  AT SEVERAL POINTS IN THE ADVENTURE YOU WILL
BE GIVEN A CHOICE BETWEEN TWO ALTERNATIVES.
  EACH TIME TYPE "A" TO CHOOSE THE FIRST
CHOICE OR TYPE "B" TO CHOOSE THE SECOND.
  [BLANK.LINE]
  PRESS THE SPACE BAR TO CONTINUE.
  END
```

The instructions for the adventure game could be printed on the screen by typing:

```
WRITE "INSTRUCTIONS
```

with the following results:

```
      ADVENTURE STORY INSTRUCTIONS

AT SEVERAL POINTS IN THE ADVENTURE YOU WILL
BE GIVEN A CHOICE BETWEEN TWO ALTERNATIVES.
EACH TIME TYPE "A" TO CHOOSE THE FIRST CHOICE
OR TYPE "B" TO CHOOSE THE SECOND.

PRESS THE SPACE BAR TO CONTINUE.
```

COMMANDS USED WITH WRITE

The TAB command tabbed the title over six spaces, and the BLANK.LINE command inserted blank lines at the indicated points. The command for a blank line is simply:

```
TO BLANK.LINE
   PRINT []
   END
```

The TAB command consists of the following procedure in Terrapin and LCSI Logo respectively.

```
TO TAB :SPACES
   REPEAT :SPACES [PRINT1 CHAR 32]
   END

TO TAB :SPACES
   REPEAT :SPACES [TYPE CHAR 32]
   END
```

COUNT

The TAB command was used to center the title of the instructions. Since there were 40 spaces across the screen, I counted the number of letters in the title (26) and subtracted them from 40:

40 spaces on the screen - 26 letters = 12

Microworlds continued

The answer (12) is divided by two to determine how many spaces should be on either side of the centered title. While I was laboriously counting letters (one .. two .. three ...), it occurred to me that I could use the Logo command COUNT to count the letters for me. Therefore I quickly typed:

```
PRINT COUNT [ADVENTURE STORY INSTRUCTIONS]
```

and obtained the result:

```
3
```

There were clearly more than three letters in the title. The command COUNT was counting the number of words in the list, instead of the number of letters in the list. After thinking for a moment, I tried:

```
PRINT COUNT "ADVENTURE
```

and got the answer:

```
  9
```

More thought. The command COUNT counts the number of words in a list, and the number of letters in a word. (Keep in mind that, in Logo, a list is a series of items enclosed in brackets, and a word is a series of letters preceded by a quotation mark.)

If you are using Terrapin Logo, note that version 1 does not have a COUNT primitive. The following procedures will provide this capability. Version 2.0 of Terrapin Logo for the Apple has a COUNT primitive.

```
  TO COUNT :ITEMS
     OUTPUT COUNT.LOOP :ITEMS 0
     END

  TO COUNT.LOOP :ITEMS :NUMBER
     IF :ITEMS = [] OUTPUT :NUMBER
     OUTPUT    :COUNT.LOOP   BUTFIRST    :ITEMS
:NUMBER + 1
 END
```

COUNTING STRATEGY

I wanted a procedure to count the number of letters in a list. Here is one possible strategy.

```
To count letters in a list
    Count the letters in the first word in
the list
    Count the letters in the next word in the
list & add the result to the number of
letters in the first word
    If all the words in the list are
exhausted, output the number
    End
```

This procedure can be expressed as follows for Terrapin and LCSI respectively.

```
  TO COUNT.LETTERS :LIST :NUMBER
     IF :LIST = [] OUTPUT :NUMBER
     MAKE "NUMBER COUNT FIRST :LIST + :NUMBER
     OUTPUT  COUNT.LETTERS  (BUTFIRST  :LIST)
:NUMBER
     END

  TO COUNT.LETTERS :LIST :NUMBER
     LOCAL :NUMBER
     IF EMPTYP :LIST [OUTPUT :NUMBER]
     MAKE "NUMBER COUNT FIRST :LIST + :NUMBER
     OUTPUT  COUNT.LETTERS  (BUTFIRST  :LIST)
:NUMBER
     END
```

COUNTING BUGS

The COUNT.LETTERS procedure has a bug in it. The number of letters in the title is 26, but the COUNT.LETTERS procedure said there were only 24 letters.

```
PRINT   COUNT.LETTERS   [ADVENTURE   STORY
INSTRUCTIONS] 0

24
```

Aha! When I looked at the English description of the procedure, I found that I was counting the number of letters in each word, but I was not counting the number of <u>spaces</u> between the words. Can you find a way to modify the procedure to fix the bug?

****

Pause for Logo detective work

****

If you are still with me, I assume that you have found a way to fix the COUNT.LETTERS procedure so that it counts the spaces between the words. My first attempt to fix the bug was to add one to each word in the title. However, that fix resulted in yet another bug, which made the count one more then it should have been. (The last word in the title doesn't have a space after it.)

more Microworlds

CENTERING

I now had a working procedure to count the numbers of letters in the title. At this point, I was a small step away from a procedure to automatically center a title. The formula for centering a title by hand was:

(40 Spaces on the Screen/Letters in Title)/2

A COUNT.LETTERS procedure can be used to write this as:

```
(40   - COUNT.LETTERS   [ADVENTURE   STORY
INSTRUCTIONS] 0 ) / 2
```

Therefore a CENTER procedure can be written as:

```
    TO CENTER :TITLE
          TAB (40 - COUNT.LETTERS :TITLE 0)/2
          PRINT :TITLE
          END
```

There is a great deal of pleasure in watching a title go zinging over to the middle of the screen.

```
CENTER [A SAMPLE TITLE]

                 A SAMPLE TITLE
```

The CENTER command can be used in conjunction with the WRITE procedure.

```
TO INSTRUCTIONS
  [ CENTER [ADVENTURE STORY INSTRUCTIONS] ]
  [BLANK.LINE]
  AT SEVERAL POINTS IN THE ADVENTURE YOU WILL
BE GIVEN A CHOICE BETWEEN TWO ALTERNATIVES.
   ...
   END
```

The CENTER command will work with both text printed on the screen, and with text sent to the printer. If the width of the screen or page changes from 40 columns to another width, adjust the CENTER procedure by changing the number 40 to the new width of the page.

Good luck, and have fun with your new centering capability. I would be interested in hearing what you do with it.

Glen Bull is a professor in the University of Virginia's School of Education, and teaches Logo courses at both the graduate and undergraduate level.

Sweet P is a Sweetheart

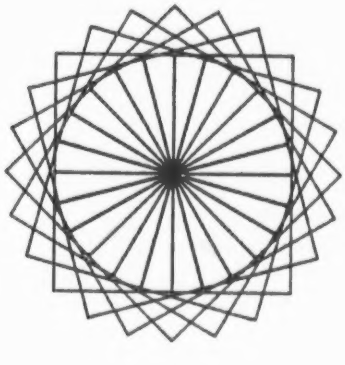
by

Steve Tipps

Among the peripherals which can be used with Logo, my personal ranking is as follows. A printer is mandatory, a robot turtle is necessary, and a plotter is desirable. But, do not underestimate the potential represented by a plotter with Logo.

There is a reason why there are not many plotters being used with Logo today. The cost of plotters has been so high that purchasing one which was both accurate and able to draw in a variety of colors was virtually impossible. Radio Shack and Commodore have had inexpensive plotters which were very serviceable. But they couldn't match some features of the big boys such as the Bausch and Lomb or Houston Instruments plotters, with price tags of $1500 and up.

One of the first moderate priced, versatile printers available is the Sweet P by Enter Corporation. Terrapin has written a set of plotter routines to interface with the Sweet P, which result in some fantastic turtle drawings.



PLOTTING WITH PLOGO

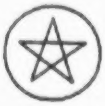
After trying several of the demonstration programs supplied with the Sweet P, software, I moved on to the Logo interface written by Mark Eckenwiler of Terrapin. The primitive Logo commands have been put into a shell (turtly speaking) of plotter commands which are named the same with the prefix P: FD becomes PFD, BK is PBK, etc.

Sweet P continued

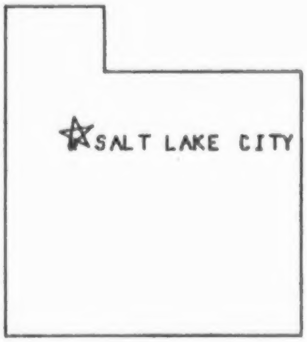
This convention seemed very natural and made converting regular turtle drawing to plotter programs quite easy. For example, CIRCLE became PCIRCLE.

```
TO CIRCLE
REPEAT 36 [ FD 30 RT 10 ]
END

TO PCIRCLE
REPEAT 36 [ PFD 30 PRT 10 ]
END
```



Commands for printing text, changing the size and direction of text, putting the pen up and down, and placing the turtle at any X or Y position and heading form a complete set of plotter primitives.

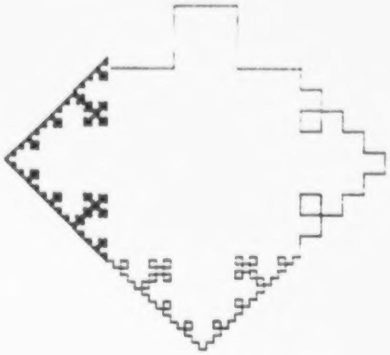


A HELP procedure provides an overview of the software and a SETUP is used to tell the computer which expansion slot is being used for the plotter.

I created a problem for myself with the primitives when I converted my standard MOVE procedure to PMOVE. This act wiped out the Terrapin PMOVE primitive, which is critical in many other commands! After a 30 minute puzzle, I discovered my mistake and made MOVEP for my MOVE.

ON LINE TEST

I worked with the Sweet P for several weeks and invited Tom Lough and Glen Bull to put it through its paces. Tom immediately went to his fractal programs (See "Fractal Fun with Logo," NLX, October 1982.) to see how accurately it would plot. He was amazed at the small closure error after such a complex drawing.



Next, I decided that the plotter needed to face the real world test. I took the Sweet P plotter to a Saturday Logo class of sixth grade students at Jack Jouett Middle School in Charlottesville. The students were working with Logo in preparation for being laboratory assistants for their teacher, Becky Poplin.

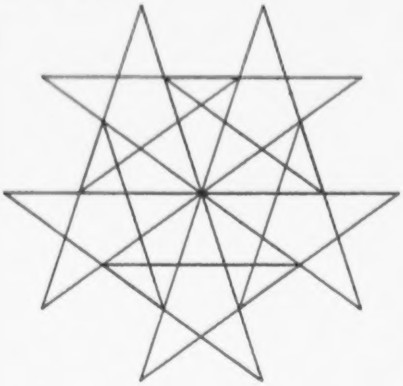
A number of them had had Logo experience the year before as part of the Albemarle County - University of Virginia Logo Project.

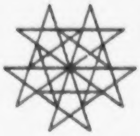
I showed them how I had made a star in a circle and the two separate programs - one for the monitor and one for the plotter. They were able to see immediately how the program was changed. When the students completed and saved their programs, they would bring their disks over to the computer with the plotter and convert them to plotter routines. I played keyboardist in this conversion process.

Watching the plotter plot was as fascinating to them as to me. They wanted to know how the plotter worked and observed carefully how the paper movement and pen movement coordinated to draw figures. Plotting out their pictures proved to be an additional incentive to make elaborate designs. I also took the conversion opportunity to work on modularity and readability of programs.

Sweet P continued

One of the nicest features we found in plotting students' pictures was the ability to change the sizes of the pictures with an expansion variable. MAKE "EXPANSION 3 is the default, but "EXPANSION value of 1 makes very good drawings one third smaller. Playing with ratios on the plotter would be an important exploration.





Two of the plotter logistics which I had to work out were threading the paper into the plotter accurately and changing the pens. Since the plotter pen senses where it is on the paper, getting it set up accurately is very important.

The manual with the Sweet P outlines a paper loading procedure that does not work because there is no "paper load button" as the manual claims. After a few attempts, I devised a system using the LOWER LEFT button which worked very well.

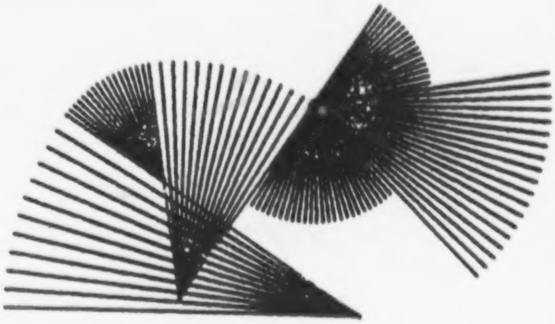
Changing pens is not difficult, but the lines will come out jiggly and wiggly unless the pen is firmly seated in the holder and the holder in the plotter arm.

After receiving specific instructions and cautions, students were able to feed the paper and change the pens without difficulty. However, changing the pens manually was a little irritating. CHANGEPEN makes the plotter pause for pen changes, but I finally got tired of taking the holder out, changing pens, and replacing it.

I didn't solve the issue but I learned to avoid it by plotting in only one color. Having enough pen holders for all the pens would be a good idea. Holders for commercial felt tip pens are available according to the user manual.

The paper which came with the plotter was fairly heavy and slick. I substituted 20 pound printer paper and it worked well, although the ink tended to soak through. Some of our designs, such as SPIN.PSQUARE, ate a hole in the paper at HOME (after 200 repetitions). We could have put the the pen up and down right at HOME or reduced the number of rotations.

I also found that turning the plotter so that its orientation was the same as the monitor was important in following and debugging programs.



SUMMARY

I found no operational faults with the Sweet P and many strengths. It costs less than $600 and should have many uses beyond Logo in schools. Overhead projections for classroom and charts for the teachers or administrators will be easy to make with the Sweet P, for example. An optional graphics program called Sweet-Plot 80 is available from Enter.

The plotter interface will be available on the Utility disk to purchasers of the new 2.0 Terrapin Logo version. Terrapin 2.0 upgrades will not include the new Utility Disk. However, you will be able to obtain the plotter program if you contact Terrapin.

Contact Enter Computer, Inc., 6867 Nancy Ridge Drive, San Diego CA 92121, (619) 450-0601 for more information on the Sweet P plotter. Terrapin Inc., 380 Green Street, Cambridge MA 02139, (617) 492-8816 can provide information on the interface with Terrapin Logo.

Steve Tipps is a regular columnist for The National Logo Exchange, and frequently contributes additional articles.

From the NLX Mailbag

Dear Editor,

I can't shake the feeling that the early introduction of recursion when teaching or presenting Logo is a classic case of dressing up a shortcoming and calling it an advantage. Let me explain.

Logo, for all its good features, has a very primitive set of operators for handling iterative processing. REPEAT gives an introduction to iteration, but the nature of iterative processing is better presented with WHILE and UNTIL operators. The ones I use with MIT Logo look like this:

```
     TO WHILE :CONDITION :ACTION
        WHILE1:   IF   (RUN  :CONDITION)  RUN
:ACTION ELSE STOP
        GO "WHILE1
        END

     TO UNTIL :CONDITION :ACTION
        UNTIL1: RUN :ACTION
        IF  NOT (RUN :CONDITION)  GO  "UNTIL1
ELSE STOP
        END
```

Once you recover from the shock of seeing a GO operator, you'll notice that the functions are, in fact, well structured. It's interesting to note that Harold Abelson in Logo for the Apple II has a similar WHILE function, but uses recursion to do it.

My feeling is that it is easier to learn the use of recursion if one has a solid grounding in iteration controlled by variables.

As a sidelight, I set up the following program:

```
     TO TEST
        MAKE "LEN 10
        WHILE [LEN < 50] [FD :LEN RT 20  MAKE
"LEN :LEN + 10]
        PRINT :LEN
        END
```

The final PRINT command will show that LEN is 50. This means that the variables within the WHILE statement are accessed by reference rather than value. This is extremely valuable, but not well documented.

John B. Boddie
Newark DE

Logo Notes

The Logo Users' Group of the Boston Computer Society has plans to resume its Polyspiral publication. For more information, write to Nola Sheffer, Co-Director, Logo Users' Group, Boston Computer Society, 8 Garden Court, Cambridge MA 02138.

Logo is gaining momentum in the Netherlands. Logo Centrum Nederland publishes an extremely interesting newsletter called Schildpad Nieuws (Dutch for Turtle News). Editor Harry Pinxteren and others report on exciting Logo happenings, along with translations of selected parts of the YPLA's Turtle News, and The Turtle Sourcebook.

Harry and his group are working on an enhanced Logo version with more LISP features and an improved system dialog. Additionally, it is being designed to be completely portable and machine independent. Periodic progress reports will be published.

If you are interested in contacting them, write Logo Centrum Nederland, Attn: Harry Pinxteren, Postbus 1408, 6501 BK Nijmegen, The Netherlands. Their telephone number is 080-238074.

Do you use Logo with handicapped children? Prentice-Hall has available a book entitled, "Making Toys for Handicapped Children," which includes details for the construction of many toys which could be used for offcomputer activities. For more information, write Prentice-Hall Inc., Englewood Cliffs NJ 07632.

Apple Computer has released the Macintosh, its newest offering. Already, rumors have begun about a Logo version for it. Digital Research Inc. (DRI), producer of DR Logo for the IBM PC, is said to have begun work on a Logo for the Macintosh.

Have you been searching for a source of "odds 'n' ends" for offcomputer activities? Send off for a copy of the "Surplus School Stuff" catalog right away! This 48-page publication contains the most incredible assortment of stuff you can imagine, at fairly reasonable prices. Send 50c to Jerryco, 601 Linden Place, Evanston IL 60202.

Logo Notes continued

Here are some more low Logo prices from Strictly Software Inc.

Commodore Logo	$57.38
PC Logo	$112.46
DR Logo	$79.96

Major credit cards are accepted. They charge $3 for shipping and insurance per order. Strictly Software Inc., 28 Barrett Road, Katonah NY 10536. (914) 232-4844.

Conroy - LaPointe Inc. advertises Terrapin Logo for $99, DR Logo for $69, and the Koala Touch Tablet for $85. Call (800) 547-1289.

For an interesting perspective on how the computer can cure our education ills, read the article about B. F. Skinner in the February 6, 1984, issue of Infoworld.

The famed Lamplighter School announces "Fun Days," a summer experience for children who have completed kindergarten. Logo computer programming will be introduced by some of the most experienced Logo teachers in the world. In addition, sessions entitled, "Logo for Teachers," will be taught for both beginners and advanced Logo users, in which educators will become acquainted with effective Logo teaching strategies and techniques. For more information, write The Lamplighter School Summer Program, 11611 Inwood Road, Dallas TX 75229.

Dr. Eleanor Roberts announces a series of Logo classes for parents and children. For more information, write to her in care of Micro-Class, 103 Marshall Bridge Road, Kennett Square PA 19348.

The University of Missouri - Columbia announces a 2-hour credit course in Logo for the summer of 1984. For more information, write to Summer Program, University of Missouri - Columbia, 301 Hill Hall, Columbia MO 65211.

Are you teaching any summer Logo courses or workshops? Do you offer Logo consultation services? If so, send us information so we can inform our readers.

NLXual Challenges

with Robs Muir

Mirror, Mirror

The difficulties encountered in solving a computer problem can usually be distilled into two distinct categories: difficulties in concept, and difficulties in coding. Most of the problems presented in this column to date have involved issues of concept, since this is of fundamental value in a Logo environment. The strength of turtle graphics is its ability to allow exploration of mathematical ideas without much syntactic (coding) knowledge of Logo.

But, Logo is more than merely a graphics language. It includes features present in only the most modern computer languages. This month's problem involves the use of some of the most important, and most overlooked, abilities of the Logo language, abilities which link Logo to the realm of artificial intelligence.

Logo has the ability to modify its own procedures, and to write new ones. Languages which permit computer programs to write more computer programs are important tools for education. Armed with such languages, we can simulate and study the very thing our minds do.

Let's take an example. If you enter the Logo editor and write a procedure to draw a rectangle, it might look something like this:

```
TO RECTANGLE :WIDTH :LENGTH
  REPEAT 2 [FD :WIDTH RT 90 FD :LENGTH RT 90]
  END
```

If you now wanted a procedure which would draw a mirror image RECTANGLE, you would have to either move the turtle to a new heading and position, or write a new RECTANGLE procedure and substitute all RT commands for LT commands. If you intend to draw many symmetrical patterns, this can be a tedious process.

Challenges continued

Fortunately, Logo can permit you to write a procedure which will write new procedures. Our problem, then, is to write a procedure which will take another procedure as input and write a new procedure with the commands RT and LT interchanged. This newly created procedure can then be used to draw a symmetrical shape.

The tools you will need for this project are the Logo primitives TEXT and DEFINE. Here are some examples of their function.

```
    PRINT TEXT "RECTANGLE
    [:WIDTH  :LENGTH][REPEAT 2 [FD :WIDTH RT
90 FD :LENGTH RT 90]]

    DEFINE "CIRC [[:SIZE][HT][REPEAT 36  [FD
10 RT 10][ST]]

    POTS
    TO RECTANGLE :WIDTH:LENGTH
    TO CIRC :SIZE
```

Variations of these primitives appear in MIT, LCSI, and TI Logo implementations. Consult your reference manual for the minor differences.

For further study on symmetry, see "Tipps for Teachers," NLX, October 1983, and Logo: An Introduction, by J. Dale Burnett, Creative Computing Press, 1983.

As with all NLXual Challenges, if you want to receive a possible solution, send a copy of your solution (or your attempt) with a letter size self addressed stamped envelope to NLXual Challenges, Attn: Robs Muir, 1688 Denver Avenue, Claremont CA 91711. We will compile contributions and send you the most elegant solution.

If you would like to submit a challenge or puzzle for this column, send your ideas to the above address.

Robs Muir is a teacher in the Claremont CA Unified School District and Claremont Graduate School. He presents Logo workshops in the southern California area.

Spanish Logo From TI Available

Texas Instruments announces the availability of TI Logo in a Spanish version for use with its TI 99/4 and 99/4A home computer systems. Several educators who teach in inner city schools with a large Hispanic population have expressed interest in such a Logo version. TI Spanish Logo I is available now for systems with a 32K memory expansion, disk drive, and controller. TI Spanish Logo II will be available in the third quarter of 1984, and requires only the 32K memory expansion unit. The price for each version is $129.95. Send check or money order payable to Texas Instruments Inc. to: Latin American Division, Texas Instruments Inc., PO Box 225621, MS 3922, Dallas TX 75265. Mark the envelope for the attention of Marge Braswell. Allow 6 weeks for delivery.

Oops

Jo Winkleman called to claim Drew Model School for Arlington (VA) schools. Steve Tipps misplaced the school in neighboring (and rival) Alexandria in his "Right Here in Turtle City" article last month. BK 20.

The National Logo Exchange, copyright 1984 by Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly, September through May, $25 (US) per year, mailed FIRST CLASS from Charlottesville, Va. $5 (US) additional per year for addresses outside the USA, Canada, and Mexico. The opinions expressed by the authors are not necessarily those of The National Logo Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress Street, Salem, MA 01970. Address other correspondence to: The National Logo Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor.....................................Tom Lough

ISSN 0734-1717

The National LOGO Exchange

Volume 2 Number 8 FORWARD 100! April 1984

Breaking Up Is Hard To Do

by
Cathy Frank

Who among us has not seen "spaghetti" programming in Logo? Sometimes it seems as if students are engaged in a contest of sorts to see who can write the longest continuous procedure!

I would like to concur with Sharon Burrowes' observations of Logo hacker habits in the January 1984 NLX. In particular, the tendency to write long procedures is distressing, especially when no thought is given to breaking them into parts. I feel that the concept of breaking problems up into smaller steps is one of the more important concepts that Logo can teach.

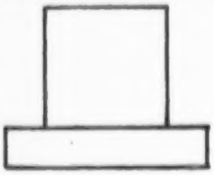
Although this concept may be obvious to us as teachers, perhaps we are not taking the time to introduce it to our students carefully enough.

I have been teaching Logo to 4th, 5th, and 6th graders for two years, and am still frustrated with their inability to grasp this concept sooner. The younger the student, the more difficult it seems to be.

As a result, I have been experimenting with various techniques as I first introduce children to Logo. Some of them are proving successful, and I would like to share them with you.

SEE THE PIECES

In order to write modular procedures, one has to see a picture in modules. That is, one must be able to see both the whole picture and its constituent parts. For example, in the picture at the top of the next column, can you see both the tophat and the rectangle and square?



I draw similar pictures on the board and ask the students to identify the objects and then list all the separate shapes they can find in them. Once they get the idea, it becomes almost like a contest to see who can list the most shapes! Followup worksheet drawings are eagerly pounced upon. Students quickly progress to the point where they begin designing pictures for each other.

MAKE YOUR OWN

Next, I give the students groups of geometric shapes (circles, triangles, rectangles, squares, etc.) cut out of heavy construction paper, and challenge them to assemble composite shapes. Usually, I will start out with a "vocabulary" of one or two shapes of various sizes, and introduce more shapes later.

My students assemble the shapes in many different combinations, creating beautiful and suprising results. Often, they will want to paste the shapes together on white paper, to preserve their creations.

LOGO PIECES

Next, I set up class projects, in which students write procedures to draw the outlines of the various shapes used in their work. Since I emphasize this concept before I introduce the students to variables, we usually put an extra letter in the title as a clue to the size. For example, SBOX might be a small square, while BBOX would be a big one.

Students place all shape project procedures in the computer and save them on disks for everyone. Now they are ready to start writing procedures which use these shapes to produce pictures on the screen.

Breaking Up continued

This technique is appealing to me because it lets me control the complexity of the projects by the number of different shapes I give the children. Although a little of the spontaneity may have been taken out of the programming aspect of the project, it is still present in the original design. The enthusiasm and excitement of the children mounts as they begin to understand the power and ease of molular programming.

Another technique I have used with older students is to suggest simple projects made up of the original shapes. These might include a wagon, a flower, a person, a house, etc. Once these have been completed and saved, the students then use them to create a more complex picture of a yard or landscape.

MODULAR SQUAD

The ultimate goal in all of this is to help students discover the effectiveness and ease of treating a complex problem in a modular fashion. After all, even the most complex picture is composed of different parts.

Once my students first experience the "bottom up" assembly of a complex picture from simple shapes, they seem to be able to grasp the "top down" concept more readily.

They don't turn away from challenging projects. Instead, they eagerly search for ways to divide them into smaller parts, and away they go. Hi yo Logo!

Cathy Frank, a parent volunteer Logo teacher for 4th, 5th, and 6th grade children in Burlington, VT, is preparing for full time work as an educational computer consultant.

Logo Teachers Wanted

Computer Summers, Inc., one of the first summer computer camp systems to focus on Logo, will operate day camps in Los Angeles, Houston, Tampa, and in Westchester County, New York, this summer, in addition to resident camps in North Carolina, Colorado, Massachusetts, and upstate New York. If you would be interested in teaching Logo at such a camp for a reasonable salary, write David Sachs, Computer Summers, Inc., PO Box 436, South Salem, NY, 10590, or call (914) 763-8829.

From The Editor

One of the more popular events in the Olympic Games is the high jump. It is always spectacular to see an athlete leap over a bar suspended far overhead. For years, high jumpers used a technique called the western roll, in which the athletes "rolled" over the bar face down.

In 1968, a young man named Dick Fosbury won the gold medal for the high jump in the Olympics in Mexico City. He astounded the world by soaring to a new Olympic record of 7 feet 4.25 inches with a revolutionary style, now called the "Fosbury Flop."

This young man ran parallel to the bar and suddenly leaped up at it backwards, sailing over it upside down! He had sought and found a better way.

One of Papert's criticisms of computer use in education is linked to resistance to change. He speaks of the so-called QWERTY phenomenon, based on the historical development of the typewriter (and computer) keyboard layout. Long after the reason for its design had vanished, the keyboard has remained unchanged. Several more efficient designs have been suggested, but, as of now, none have caught on. We've always done it this way.

Perhaps it is time to take a look around our Logo circles to see if similar signs of stagnation are setting in. Is this happening with our Logo teaching? Is it happening with Logo itself?

Many have found certain teaching methods and techniques which seem to be effective. Are we still trying to refine them? Do we continue to search for still other ways?

Let us not think that we have found the best way yet. Rather, let us always enjoy the search for it.

After all, if we ask our students to take risks in their Logo work, we should be willing to take risks in our Logo teaching, shouldn't we?

FD 100!

TIPPS for TEACHERS

by
Steve Tipps

Unsquare Square:
The Story of Scrunch

A Logo student sits down at a computer to draw a square. But it turns out not square, but rectangular! Circles are ovals! What computer madness is this?

This squeeze of shapes results from the monitor being set to a different ratio of vertical picture elements (pixels) to horizontal pixels than the computer expects. Normally, Logo expects for the monitor to be set at about 8 steps vertical for 10 steps horizontal and automatically adjusts the commands to this ratio.

Having nonsquare squares and noncircular circles is disconcerting, but easily fixable. If the square or circle is flatter than desired, try resetting the ratio to a different value. For example, to set it to a ratio of 1 step vertical to 1 horizontal, type

```
     .ASPECT 1         (MIT versions)
     SETSCRUNCH 1      (Apple LCSI version)
     .SETSCRUNCH 1     (IBM LCSI version)
     .SETSCR 1         (Atari LCSI version)
```

The monitor may also have a switch on the back which can be reset to correct the ratio. But the command which corrects monitor deflection can can also stimulate new turtle explorations.

PROPORTIONAL MICROWORLDS

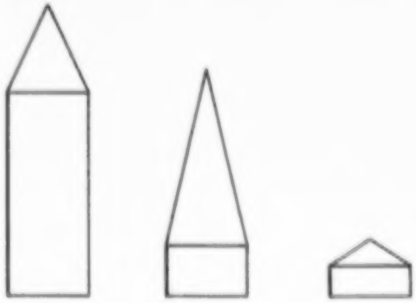
The scrunch commands provide experience with ratio and fractions. Except for Atari Logo, they can be set at levels from hundredths to hundreds, although .1 to 6 seems to be the most interesting range for initial explorations.

The effect of setting the ratio is to change the number of vertical moves to 1 horizontal move. The CIRCLE drawn at times as tall as wide; the SQUARE at SETSCRUNCH .2 is really a rectangle which is 1/5 as long on one side as it is on the other.

One way to illustrate this off the computer is to draw a shape on a piece of paper with wide lines. Fold the paper into narrow strips like a paper fan. The shape on the paper can now be scrunched up or expanded by pushing or pulling the edges of the paper like an accordion. Try this fan folding with different shapes and pictures to see what scrunch will do.

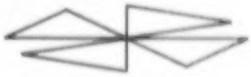
NEW SCRUNCH FOR OLD FRIENDS

Some interesting and unexpected things happen when old familiar drawings are done with different aspects. The classic Logo house becomes either a tall thin abode or a squat narrow ranch-style one. Rotating a regular figure about HOME creates interesting petals and overlaps.



Equilateral and isosceles triangles can be used for beginning explorations.

```
   SETSCRUNCH 2 TRIANGLE 30
   SETSCRUNCH 3 REPEAT 4 [TRIANGLE 25 RT 90]
   SETSCRUNCH  .5 REPEAT 9 [ FD 10  TRIANGLE
20 FD 10 RT 40 ]
```



Rotating the triangle changes how each side is drawn because the ratio changes the effective angle as well as the length. This quirk can be used to create triangles anywhere on the screen. Changing the scrunch for triangles is useful because the horizontal base can be fixed while vertical sides change.

Since the horizontal dimension is always the standard, the base of the triangle can be set with a variable and the height of the triangle set with the scrunch. A family of triangles can be made up with SETSCRUNCH or set the scrunch back to .8 before leaving the procedure so that the next shape is not likewise distorted.

Tipps continued

```
TO TRIANGLE.HT3 :SIZE (Apple LCSI version)
   RT 30
   SETSCRUNCH 3
   TRIANGLE :SIZE
   SETSCRUNCH .8
   LT 30
   END
```

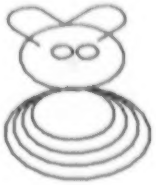
```
TO TRIANGLE.HTHALF :SIZE    (MIT version)
   RT 30
   .ASPECT .5
   REPEAT 3 [FD :SIZE RT 120]
   .ASPECT .8
   LT 30
   END
```

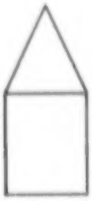
Scrunch can even be set through an input variable.

```
TO SCRUNCH.SQUARE :HEIGHT
   SETSCRUNCH :HEIGHT
   SQUARE 50
   SETSCRUNCH .8
   END
```

```
TO SCRUNCH.SQUARE :HEIGHT
   .ASPECT  :HEIGHT
   SQUARE 50
   .ASPECT  .8
   END
```

Using scrunched squares and triangles, different combinations of houses can be built for three bears made of scrunched circles.





SCRUNCH, SQUEEZE, AND FLIP

Scrunch can be used to squeeze pictures or expand them. This can be handy in particular situations when you want to slightly alter or distort a shape.

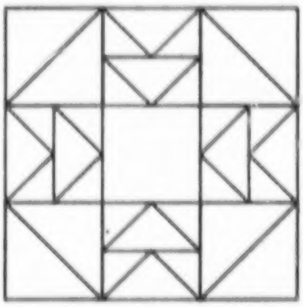
One such use for scrunch was found by two students in our Introduction to Instructional Computing class at UVA. Cindy Rutledge and Jane Blanchard wanted to draw a melting snowman. They had tried redrawing and repositioning the snowman in a recursive procedure, but it kept wrapping off the bottom of the screen. Even after that problem was solved, they were unhappy with the results.

Scrunch was a solution to their problem. The snowman got flatter and flatter until it "melted" into a flat line.

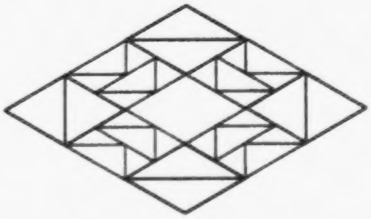
They then used scrunch in another procedure to draw a balloon which sailed into the sky. As the balloon went up, it got smaller and flatter. The perspective was tremendous. They were able to "animate" the balloon with several READPICTs using MIT Logo.

Scrunch can also be used to change the shape of quilt block designs. Working out quilt blocks with Logo is a great project in the first place. (See "Logo Quilting Party," by Tom Lough, NLX December 1982.) Scrunch allows you to change square blocks into diamond shaped blocks without calculating all the angles.

You can begin with a simple design based on squares and diagonals. After this is worked out to your satisfaction, simply turn the block on end, change the scrunch and draw it as a diamond. Becky Poplin found that a scrunch of 1.4 made a good diamond shape from quilt blocks. Here are samples of the DOUBLE-T block which she transformed into a DOUBLE-T DIAMOND.



Tipps continued



A great resource for ideas on changing square blocks to diamonds is <u>Diamond Patchwork</u>, by Jeffrey Gutcheon, $9.95, from Alchemy Press, 510 Broadway, New York, NY 10012.

Another scrunch exploration which can be done with most Logo versions (one notable exception is the IBM LCSI version) is to use negative numbers as input. In a recursive routine with scrunch, this produces an effect like a mirror, resulting in a symmetrical design as the scrunch falls through 0 to the negative numbers.

```
TO SCRUNCH.DIAMOND :HEIGHT    (MIT version)
   .ASPECT :HEIGHT
   LT 45
   SQUARE 50
   RT 45
   SCRUNCH.DIAMOND :HEIGHT - .2
   END
```

This procedure needs a test to stop, of course.

TRUTH and SCRUNCH

While scrunch is great fun and should provide a new way of exploring turtle drawing, it has one major problem — truthfulness. The computer thinks that the angles of the square are still 90 and the sides are all the same, even if the picture is a diamond. What should a teacher should do about this dilemma?

The sophistication of the students suggests different approaches. Younger students explore the shapes as they are changed without any great concern. The distortions are accepted as part of the intrinsic geometry of the turtle's world.

When descriptive geometry is introduced, distortion of the angles needs consideration. Use protractors and draw equilateral triangles on grid paper with scales of 3 to 1 or distorted.

Scrunch plays a trick on the turtle which makes it think it is still drawing equilateral triangles. If you add a PRINT HEADING to an equilateral triangle

```
REPEAT 3 [ FD :SIZE PRINT HEADING RT 120]
```

the headings remain constant regardless of the scrunch. The discrepancy between the turtle's angles and the paper angles raise new questions about the rules of the turtle world. When is a square not a square? The answer is a clue to another aspect of the turtle world.

Support from Enter Computer, Inc., 6867 Nancy Ridge Drive, San Diego, CA 92121, in producing the SWEET P plotter drawings is gratefully acknowledged.

Steve Tipps is a professor in the University of Virginia's School of Education, and presents Logo workshops for school systems throughout the United States.

Logo for the Sony

by Hillel Weintraub

Logo versions are popping up in Japan. One of the more interesting is the 8-bit version of DR Logo supplied with the Sony SMC-777 computer. When the systems disk is booted, the user can choose either BASIC or Logo. As far as I know, this is the first Logo version supplied with a computer by a manufacturer.

There are several nice features in this Logo system. If your procedure has a bug, typing ED will take you to the editor, with the cursor at the bug. The graphics and text pages overlap, allowing for full mixtures. The color of the text can be changed and flashed. You are given a choice of 4000 (four thousand) background colors!

There are no LAST and BUTLAST commands, and the graphics are fairly slow, about half the speed of comparable Apple versions.

The Sony SMC-777 computer with its particular DR Logo version is planned for release in the US in late 1984.

Hillel Weintraub is a teacher at Doshisha International High School in Kyoto-fu, Japan, and is the president of the Society for Microcomputing in Life and Education (SMILE).

Apple Sprite Logo

A review by
Griff Wigley

For well over a year, Logo Computer Systems Inc. (LCSI) has demonstrated versions of Apple Sprite Logo at various conferences, teasing the TI Logo diehards among us who yearned for a Logo that combined the best of both the Apple and TI versions. At last, it has been released. Despite some limitations, Apple Sprite Logo easily surpasses the Apple / TI combination.

Apple Sprite Logo is both a hardware and a software package. That is, it comes with both a plug in card and a language disk. The card goes into slot 2 of the Apple computer, and the monitor cable attaches directly to it. Once the card is in place, you can boot the disk to begin.

HERE COME THE SPRITES

Apple Sprite Logo comes with 30 sprites (dynamic turtles). They can take on whatever shape and color you like, and be given speed and heading that are retained while the computer attends to other matters.

In addition, they can draw, detect collisions, and leave copies of their shapes on the screen. The latter delightful feature is invoked by the command STAMP. One of my favorite STAMP projects was to write a procedure which printed a path of sprite shapes across the screen. It made a great addition to my version of Beachworld for my pre-schoolers. (See Byte Magazine, August 1982, "Designing Computer Based Microworlds," by R. Lawler.)

With the predicate TOUCHINGP, you can write procedures which determine whether a sprite is touching another, or any one of a list of others. Need I elaborate on the possibilities here?

SHADES OF TI LOGO

Since at least one of the chips on the board is made by TI, there are some direct similarities to TI Logo. For example, Apple Sprite Logo has the same maximum speed of 127, and the same "5 sprites in a row" problem. Other similarities incude:

FREEZE and THAW -- halts and releases all moving sprites.

SETXVEL and SETYVEL -- sets x- and y-components of the velocity. (Bouncing sprites are a great activity for these commands.)

EACH -- causes separate sprites to follow a list of instructions.

Colors — sprites and backgrounds can be one of 16 different colors.

BETTER THAN EVER

Sprite enhancements over TI Logo are:

ASK -- allows for momentary talking directly to one or more sprites without affecting others who are currently listening.

HIDE and SHOW — same as HT and ST.

More shapes are available — up to 56 different shapes can be defined at any one time.

MOVE -- moves a sprite at a certain speed for a certain period of time.

SLIDE — moves a sprite forward a certain number of steps in a certain period of time.

COLOR.OVER — detects the color under a sprite.

SNAP — replaces sprite's shape with the pattern on which its 16 x 16 grid is resting.

OTHER GOODIES

Apple Sprite Logo has some impressive abilities for background graphics:

LOADPICT and SAVEPICT — same as MIT Logo for saving and loading graphics.

CHANGE.COLOR -- allows for changing ink and text color.

FILL -- fills an area with the color of the sprite's pen.

SHADE -- fills an area with copies of the shape of the sprite inside that area, in the color of its pen.

IN CHARACTER

Although sprites get the publicity, I feel the character graphics really enhance the Sprite Logo scene. The character manipulation of TI Logo has been improved upon in several ways.

Character shapes can be defined, edited, and saved just like sprite shapes. For example, you can redefine the "4" to look like a "5", so that PRINT 2+2 will give 5 as an apparent result!

Sprite Logo continued

There are fewer restrictions for coloring the characters, too. Apple Sprite Logo has a command called SETTC (short for SeT Text Color) which changes the foreground and background color of each character. There are no character sets to worry about. If you want the alphabet to be different combinations of colors on the screen at the same time - no problem! Another feature I like is Sprite Logo's ability to use either upper or lower case letters with an Apple IIe.

All these graphics capabilities allow for fancy layering. The screen color is the bottom layer. Characters and graphics from STAMP, FILL, SHADE, and pen drawings share the middle. Sprites occupy the top. Additionally, sprites themselves are layered, with lower numbered sprites having visual priority, permitting a passing behind / in front effect.

READING MATTER

The Apple Sprite Logo documentation is thorough. Two introductory sections, one on turtle graphics and one on dynamic turtle graphics, contain many sample projects. The reference manual contains explanations and examples of every feature of the language, including all the word and list processing features of Apple Logo.

BUT ... BUT

Alas, Apple Sprite Logo is not perfect. My biggest wish is that it had music and speech. Now, that's not asking too much, is it? Time to dream about MacIntosh Logo...

Most of the minor problems are evidently due to hardware limitations. The one that bothered me most was the lack of a cursor indicator when scrolling text on the screen. If you type CATALOG, POTS, POALL, etc., and toggle the scrolling with CTRL-W, it is difficult to tell the new text from the old.

CTRL-S, -T, and -L are included no more. You must use the full commands, or, as I did, define your own short forms.

The turtle doesn't visually rotate as often as the Apple and TI Logo turtle. For example, if the turtle is given a series of eight RT 5 commands, you will see it pivot only three times. If you type SETHEADING 82, it appears to point the same as SETHEADING 90. This is not a major drawback, but may cause some initial confusion for beginners.

A sprite can draw while in motion with the SETSPEED command, but, as the speed increases, the line becomes increasingly dotted. With FD and BK commands, the line is solid all the way, as usual.

The negative y-coordinate boundary changes if you switch between SPLITSCREEN and FULLSCREEN.

When first calling on a sprite, you must set its shape to the turtle before it will accept another shape. Maybe that's why I should be calling them turtles instead of sprites!

I've had two crashes with the version of Apple Sprite Logo I tested. "!!!LOGO SYSTEM BUG!!!" was the provided error message, as the screen gradually went bananas. Rebooting the disk was the only way to restore things to normal. The version I have is at least nine months old, and newer versions may not have this problem.

As you can see, I have to pick nits to find fault with Sprite Logo. I can live with these problems, considering the additional Logo power available.

TO GET YOUR COPY

LCSI has elected to distribute Apple Sprite Logo, instead of using the network of Apple dealers. Each Sprite Logo set (includes card and disk) costs $299. Discounts are available for purchases of five or more units. Send your order or request for more information to Norman Baron, Logo Computer Systems Inc., 220 Fifth Avenue, Suite 1604, New York, NY 10001, or call (212) 684-0710.

WHO AND WHY

Apple Sprite Logo would be ideally suited for home and computer camp use. Although most schools would probably add peripherals such as color monitors and printers to their Logo setups before anything else, I would advise those who can afford it to put Sprite Logo at the top of their priority lists for hardware and software purchases. (It's both, remember.) It will enrich your Logo environment ten-fold!

Griff Wigley is a regular columnist for The National Logo Exchange, and frequently contributes additional articles.

Boston Logo Users' Group News
diSessa and the Dynaturtle

reported by
Nola Sheffer

At the January meeting of the Boston Logo Users' Group, MIT Physics Professor Andy diSessa reported on his work with students and the dynaturtle.

"Kids interact with a dynaturtle in a different way than they do with the regular Logo turtle," he began. With a geometry turtle, you can understand what is going on fairly quickly, even though what you are doing may have sophisticated implications.

In contrast, when working with a dynaturtle, designed to model the laws of physics as they apply to a frictionless zero gravity environment, it is difficult to understand what is happening. Our intuition is not a good guide.

For example, if asked how to force an object to travel in a circle, you might reply that you push it in a circular path with a tangential force. However, in reality, you must keep exerting an inward force.

FALSE INTUITION

These false intuitive notions of physics last a long time, and it is difficult to develop sophisticated ones. However, good intuition is important to being a good physicist; it is not enough to be able to manipulate formulae. diSessa believes that a physics microworld might help develop this much needed sophisticated intuition.

He explains that microworlds come in all intellectual domains. He uses the metaphor of an intellectual playground where children can interact in an unrestrained way. This environment allows for:

(a) pleasurable repetition of skills with which the student is comfortable,
(b) personal goals and an individual style in reaching those goals,
(c) a sense of power (reward) for feeling competent and in control, and
(d) a rich social setting.

SIMILAR STRATEGIES

In his work with kids and dynaturtles, diSessa found that the students all tended to go through a similar sequence of strategies. For example, when trying to use the turtle to hit a target, students tended to:

(1) try to aim straight at the target and kick, and
(2) move the turtle even with the target, turn 90 degrees, and kick.

Both strategies are unsuccessful.

(Note: In such a situation, the typical location of the turtle is at the bottom of the screen, with the target located up and perhaps to the right 45 degrees. The user can turn the turtle through increments of 30 degrees right or left, and apply a kick or impulse in the direction the turtle is facing.)

One successful strategy hit upon by many was that of the "antikick." This cancellation idea has been used by students at all levels, from upper elementary to college. In fact, diSessa found that first year college students tended to go through the same strategies as elementary kids. This may be a comfort to you the next time you are struggling through one of the dynaturtle games!

These strategies can be used, diSessa feels, as a means of explaining laws of physics, and helping students develop an accurate reliable intuition, rather than the immature one with which most of us operate.

I hope that the work Professor diSessa has done with the dynaturtle can be developed and expanded into a form which can be readily used by high school and upper elementary school teachers. It seems too rich a resource to lie undeveloped!

Thank you, Professor diSessa, for sharing your physics microworld.

Nola Sheffer is Co-Director of the Logo Users' Group of the Boston Computer Society.

Here are the current Logo prices from Gold Disk. Krell Logo $74.95, Terrapin Logo $119.00. Call 1-800-368-2260.

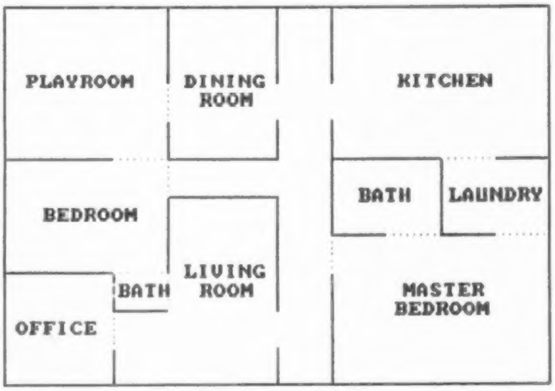
MICROWORLDS

by
Glen Bull

ETV: Educational TurtleVentures
Part I

Previous columns have dealt with using Logo to write adventure stories. Now, let's explore how to use Logo to design an adventure game!

Shown below is the floor plan of a turtle's home ... often a place of adventure.



The rooms of this turtle's home are conveniently labeled, but that is not always the case. During the course of an adventure, it is sometimes necessary to ask questions, such as:

```
TELL WHERE?
DINING ROOM

TELL WHAT?
CANDLE KEY MIRROR
```

These commands help determine the room in which the turtle is located, and what is in the room. Along with procedures which will enable the turtle to CARRY and DROP objects, these will form the main part of our adventure game design. Let's begin with an exploration of how to develop TELL WHERE?.

WHERE IS THE TURTLE?

There are a number of ways to determine the room the turtle is in. The method I will develop has three advantages.

(1) It requires only a few short procedures.

(2) It can locate the turtle quickly.

(3) It can easily be adapted to rooms of any size or shape, as long as the corners of the room are square.

It has the disadvantage of being somewhat wasteful of memory. Up to 100 variables are used to follow the turtle from room to room.

Think of the screen as a grid of blocks, numbered 0 through 99.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

Each room is defined by the blocks which fill the room. For example, the following blocks make up the office in the lower left hand corner of the turtle's home:

70 71
80 81
90 91

Thus, if we can find the block the turtle is on, we can tell the room quickly.

Logo has coordinate commands which tell how many steps the turtle is from the center of the screen. If the turtle is 70 steps to the right of the center of the screen,

```
PRINT XCOR
```

will give this result:

```
70
```

The screen I am using at this time is 320 turtle steps wide. (The number of turtle steps on your screen may be different depending on the brand of computer you are using.) Suppose that I want to divide the screen into ten blocks across, as in the above diagram. Each block will be 32 turtle steps wide.

Microworlds continued

320 (screen width) / 10 (number of blocks) =

32 (width of each block)

Suppose the turtle is in the top row, on the seventh block from the left side of the screen. Can I identify this block by dividing the turtle's position by the width of a block?

```
PRINT XCOR / 32
```

When I tried this, I was surprised to get the answer:

```
2.1875
```

SHIFTING TURTLES

The turtle was trying to tell me that it was on block 2, not block 7. A moment's thought supplied the answer. I was thinking of the screen in this way:

1 2 3 4 .... 318 319 320

The turtle was thinking of the screen in this way:

-159 -158 -157 ... 0 ... 158 159 160

In my mind, I thought of the turtle's position as running from left to right. In reality, the turtle's x-coordinates are positive to the right of center, and negative to the left of center. How can the turtle's postition be changed to all positive numbers? Add 159.

```
PRINT (XCOR + 159) / 32
7.15625
```

Now the turtle correctly told me that it was on the seventh block across the screen. The value added to XCOR to make all the numbers positive is called "offset". Therefore the number 159 could be called the horizontal offset.

The turtle told me more than I wanted to know, since I didn't care about the fractional part of the number. Asking for the integer of the result took care of this problem.

```
PRINT INT (XCOR + 159) / 32
7
```

BLOCKING THINGS OUT

Now let's write a procedure to tell us what block the turtle is on. A setup procedure defines the variables which will be needed. Use the number of turtle steps for the height and width of the screen in your particular version of Logo in SCREEN.HEIGHT and SCREEN.WIDTH. The number of blocks in each row and column can be any number you wish. I will use 10 here, for a total of 100 blocks. The horizontal offset is used to change the blocks from this:

-4 -3 -2 -1 0 +1 +2 +3 +4 +5

to this:

0 1 2 3 4 5 6 7 8 9

```
     TO SETUP
       MAKE "SCREEN.WIDTH  320
       MAKE "SCREEN.HEIGHT 250
       MAKE "NO.OF.ROWS 10
       MAKE "NO.OF.COLS 10
       MAKE "BLOCK.WIDTH  :SCREEN.WIDTH  /
:NO.OF.COLS
       MAKE "BLOCK.HEIGHT :SCREEN.HEIGHT /
:NO.OF.ROWS
       MAKE "HORIZONTAL.OFFSET (:SCREEN.WIDTH
/ 2)  - 1
       MAKE "VERTICAL.OFFSET (:SCREEN.HEIGHT
/ 2) - 1
       END
```

After the SETUP procedure is run, we can use the following procedure to obtain information about the column and row of the block the turtle is on.

```
    TO GRID.LOCATION
      LOCAL "X LOCAL "Y
      MAKE "X INT (XCOR + :HORIZONTAL.OFFSET)
/ :BLOCK.WIDTH
      MAKE "Y INT (YCOR + :VERTICAL.OFFSET)
/ :BLOCK.HEIGHT
      MAKE "Y (:NO.OF.ROWS - 1) - :Y
      OUTPUT LIST :X :Y
      END
```

For example, if the turtle is on block 3 of the top row,

```
PRINT GRID.LOCATION
3 0
```

The top row of blocks is row 0. Remember that the numbering starts with 0, so that the fourth block over would be labeled 3:

	Block in Row 1st	2nd	3rd	4th
Column on Grid:	0	1	2	3

The only Logo command in this procedure which may be unfamiliar is OUTPUT. The command OUTPUT deserves a column or two of its own, but I have never felt confident enough to write one. (Any volunteers out there?)

Briefly, OUTPUT allows a procedure to act as a function. A function may be thought of as a procedure which accepts an input and returns a value. In this case, the input is the turtle's coordinates, and the output is the column and row information about the block the turtle is sitting on.

What does the next to last line (MAKE "Y (NO.OF.COLS -1)-:Y) of the GRID.LOCATION procedure do? Try removing this line and see what happens.

BLOCKING THINGS IN

The blocks in the the sample grid are individually numbered. For example, block 5 in row 3 is labeled 35. If the turtle is placed on that block,

```
PRINT GRID.LOCATION
```

produces the result 5 3, and not 53, or 35. Although it might simplify things to number the grid differently, let's use what we might remember of list processing to combine these two numbers into the block number.

WORD AND LOCATION

The procedure GRID.LOCATION outputs a list of two numbers: the column and the row the turtle is sitting on. The list commands FIRST, LAST, and WORD can be used to combine these two numbers into a single number.

The Logo command WORD is used to glue two words together:

```
PRINT WORD "FIRE "TRUCK
FIRETRUCK
```

It can also be used to glue numbers together.

```
PRINT WORD  5 3
53
```

The command WORD can be combined with the commands FIRST and LAST to glue the first and last members of a list together in reverse order.

```
PRINT FIRST [5 3]
5
PRINT LAST  [5 3]
3
PRINT WORD LAST [5 3] FIRST [5 3]
35
```

(Actually the definition of a "word" in Logo is slightly different from the English definition of a word, and can include numbers. A Logo "word" is any item with a quotation in front of it. In most informal Logo versions, the quotations are optional in front of numbers.)

Now we can create a procedure which tells the exact block the turtle is sitting on.

TURTLE TREASURES

In the meantime, try burying treasure under some of the blocks by typing a description of the treasure after the name of the block.

```
MAKE 37 [DIAMONDS]
MAKE 14 [GOLD]
```

(Note: With some MIT Logo versions, a quote is needed before the number. For example, MAKE "37 [DIAMONDS].)

To find out if a block with a particular name (for instance, block 37) has treasure buried under it, type:

```
PRINT NAMEP 37
```

(Note: Some MIT versions require a quote before the number, as in PRINT THING? "37.)

```
TO BLOCK?
   LOCAL "XY
   MAKE  "XY GRID.LOCATION
   OUTPUT WORD LAST :XY FIRST :XY
   END
```

Note that this procedure will work only if there are no more than 10 rows or 10 columns of blocks.

```
PRINT BLOCK?
35
```

In next month's column the command WHERE? will be developed. WHERE? tells the name of the room the turtle is in by referring to a list of blocks associated with the room.

Microworlds concluded

Logo will print TRUE if there is a buried treasure under the block, and FALSE if there is not. If there is treasure present, you can identify it by typing:

```
PRINT THING 37
```

In this case, Logo will respond with:

```
DIAMONDS
```

Here is a procedure to tell if there is treasure on the block where the turtle is located.

```
    TO TREASURE?
       TEST NAMEP BLOCK?
       IFTRUE OUTPUT THING BLOCK?
       IFFALSE  OUTPUT [THERE IS NO TREASURE
HERE.]
       END
```

After burying treasures under some of the squares, use the procedure by typing:

```
PRINT TREASURE?
```

If the turtle is sitting on a block with treasure under it (such as block 14), it will reply:

```
GOLD
```

Young children can use the following procedure to hide treasure at any turtle location they choose, without having to worry about the block number.

```
TO HIDE :TREASURE
     MAKE BLOCK? :TREASURE
     END
```

For example, while the turtle is in the office, HIDE [MONEY] would hide MONEY beneath whatever block the turtle is on.

To make a treasure hunt more interesting, clues could be placed under the blocks also.

```
MAKE 41 [NORTHEAST TO SHINY YELLOW]
```

If the turtle is on block 41 when you type

```
PRINT TREASURE?
```

Logo will respond with

```
NORTHEAST TO SHINY YELLOW
```

This may help you find the gold at block 14.

AN INVITATION

There are many other applications which might make use of the BLOCK? and TREASURE? procedures we have developed. For example, you might attach crosses and circles to the blocks on a 3 by 3 tic tac toe grid. I substituted the coordinates from a touch tablet (the Koala Pad) in place of the turtle's screen coordinates in the GRID.LOCATION procedure and associated turtle commands with each region on a 5 by 5 block Koala Pad grid to create an Instant Touch Logo.

What other uses can you think of for BLOCK? and TREASURE? We will offer a prize for the best suggestion. Send your ideas to Glen Bull, NLX Microworlds, PO Box 5341, Charlottesville, VA 22905, before May 15, 1984.

Note for MIT Logo Users

The procedures developed in the column for SETUP, GRID.LOCATION, BLOCK?, TREASURE?, and HIDE are provided in the LCSI version of Logo for Apple and IBM PC. If you are using Terrapin or Krell Logo for the Apple, or Terrapin Logo for the Commodore, there are slight differences for some procedures.

```
TO SETUP - no change

TO GRID.LOCATION
  MAKE "X INTEGER (XCOR + :HORIZONTAL.OFFSET)
/ :BLOCK.WIDTH
  MAKE  "Y INTEGER (YCOR +  :VERTICAL.OFFSET)
/ :BLOCK.HEIGHT
  MAKE "Y (:NO.OF.COLS - 1) - :Y
  OUTPUT LIST :X :Y
  END

TO BLOCK?
  MAKE  "X.Y GRID.LOCATION
  OUTPUT WORD LAST :X.Y FIRST :X.Y
  END

TO TREASURE?
  TEST THING? BLOCK?
  IFTRUE OUTPUT THING BLOCK?
  IFFALSE OUTPUT [THERE IS NO TREASURE HERE.]
  END

TO HIDE - no change
```

Glen Bull is a professor in the University of Virginia's School of Education, and teaches Logo courses at both the graduate and undergraduate level.

NLXionary

A lectionary of selected Logo readings

compiled by Griff Wigley

Welcome to the NLXionary! Each month, we will present a lectionary of articles on Logo from other publications for your reading pleasure.

RECENT

PCjr Magazine, April 1984, "The Turtle as Tutor," by James Chumbly.

>An overly enthusiastic report of Logo's impact on the author's 9 year old daughter, within the framework of a brief introduction to IBM Logo. Though it seems innocent enough (a newcomer's excitement), articles such as this may build unrealistic expectations in parents for their children's first exposure to Logo.

PCjr Magazine, April 1984, "The Advanced Course," by Winn L. Rosch.

>This fellow knows Logo and writes with wit to boot. History, philosophy, and applications for Logo all are given interesting treatment here. His "Logo is better" arguments are concise and unusual. Inserts include an example of a neat Logo business graphics program, and a disagreement with his editors on LOGO vs. Logo. Don't be misled by the ridiculous pictures of adults dressed up in turtle suits which grace the cover, the two Logo articles, and the editor's column. They undermine Rosch, who argues convincingly for taking Logo seriously.

Compute!, November 1983, "Bucky and the Turtle," by David Thornburg.

>More stimulating thoughts on turtle geometry. Thornburg connects Buckminster Fuller's futuristic Synergistics philosophy - a "geometry of thinking" - with the mathematical formalism of turtle geometry.

Teaching, Learning, Computing, November 1983, "Papert Reflects on Logo, Past and Future," by Barbara Kellam-Scott.

>A well-written account of an interview with Papert. His recent thinking matches that set forth in "Mindstorms," but he's more pragmatic about the extent to which schools can accommodate total self-directed learning environments.

TurtleTalk, Winter 1984, "The Future of Logo," by Edward and Mark Friedland.

>A more accurate title would be "The Thinking Behind the Features of Our New Logo." The authors describe SOCRATES LOGO (caps are theirs), currently being developed by Krell Software. Multiple touch sensitive turtles, voice input, robot output, redefinable error messages, and a tutorial package are all promised. Their rationale about how current versions of Logo don't go far enough in achieving Papert's purposes is quite interesting, yet the promotional tones of the article overrides and obscures their message. Nonetheless, it sounds like a version to get excited about. It is apparently scheduled for release late this year.

ANCIENT GEM

Byte Magazine, August 1982, "Why Logo," by Brian Harvey.

>Track this article down if you have been involved with computers and Logo for a while now, and are yearning for a deeper understanding of Logo, both as a computer language and a philosophy. This piece is tough sledding at times; yet, every time I re-read it, I learn something new. Harvey thoroughly covers the Logo waterfront in this article, and virtually none of it is out of date. He is one of a rare breed who can talk about subjects over your head in an interesting way, thus teasing you back again and again.

MENTIONABLES

Educational Technology, January 1984, Ed Tech Product Reviews, "Print Materials on Learning Logo for Children and their Teachers," by Douglas Clements.

>A thorough four page review of three Logo products for the classroom: "Logo: A Problem Solving Approach," a set of activity cards by Joan Davis and Joan Schwenker, "1,2,3, My Computer and Me," a book by Donna Bearden, and Dan Watt's "Learning with Logo." You will find no surprises here.

The Christian Science Monitor, January 27, 1984, "Learning a Computer Language May Be Easier Than You Think," by Lois Colt.

Griff Wigley is co-owner of Family Computing, Inc., a facilitator in the Faribault (MN) Public Schools, and a school board member of Prairie Creek Community School.

ParaboLogo

by
Tom Lough

From the beginning, I was intrigued with Logo's ability to draw complex geometric shapes without using complex algebraic formulae. The circle is a classic example, of course.

```
REPEAT 360 [FD 1 RT 1]
```

This expression is both simple and elegant. It is easily understood in terms of how we move ourselves. If we go repeatedly forward 1 step and turn right a little, we will walk in a circle.

I have often wondered, "What about a parabola?" Is there a simple and elegant way to draw parabolic paths without using an algebraic relationship?

Once again, the key is in how we would move ourselves. But who moves in a path that looks like a parabola? We do, when we jump from the bed of a truck or a porch. Our path through the air is a combination of horizontal movement at a nearly constant speed and downward movement accelerated by gravity.

How can this be put together in a Logo procedure? One way is to do it in parts.

The part with the movement at constant speed can be modeled by the following procedure.

```
TO CONSTANT :SPEED
     FD :SPEED
     CONSTANT :SPEED
     END
```

Experimenting with commands such as CONSTANT 10, CONSTANT 1, and CONSTANT .3 will result in turtle movement across the screen at constant speeds. The endless recursion can be stopped with CTRL-G or CTRL BREAK for most Logo versions.

The part of the movement with acceleration is a little more difficult. Acceleration increases the distance traveled in each increment of time. Constant acceleration, such as that of earth's gravity, means that the extra distance traveled in each increment of time is the same.

For example, consider this pattern of dots marking the position of a body after equal intervals of time. The body here is moving from left to right, with a constant acceleration.

. .   .   .

The distance between the dots increases as the body moves from left to right. If you will measure the distances between adjacent dots, you will find that the <u>difference</u> between them is a constant amount.

Thus, a sequence of commands like

```
FD 3 FD 5 FD 7 FD 9 FD 11 FD 13 FD 15
```

would result in a brief accelerated motion of the turtle.

Again, a recursive procedure is an ideal way to simplify the concept.

```
TO ACCELERATE :DISTANCE :EXTRA
  FD :DISTANCE
  ACCELERATE (:DISTANCE + :EXTRA) :EXTRA
  END
```

An enhancement I suggest to my physics students is for the turtle to mark its successive positions with dots. With LCSI versions, the command DOT POS works nicely. With MIT versions, it is necessary to define a DOT procedure. A popular one is

```
TO DOT
  PD FD 1 BK 1 PU
  END
```

Thus, the ACCELERATE procedure becomes

```
TO ACCELERATE :DISTANCE :EXTRA
  DOT
  FD :DISTANCE
  ACCELERATE (:DISTANCE + :EXTRA) :EXTRA
  END
```

** Note: If you have Apple (LCSI) Logo, use PD DOT POS PU instead of DOT in the above procedure. For IBM (LCSI) Logo, use DOT POS.

Experimentation with ACCELERATE and various combinations of input numbers will produce a pattern of dots similar to those above.

Now let's put the horizontal and vertical motions of our jump together in an attempt to trace a parabolic path.

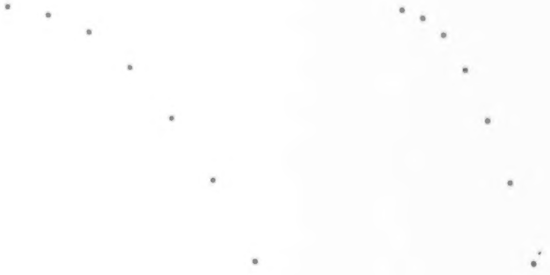
ParaboLogo continued

```
TO JUMP :HOR :VERT :GRAVITY
   DOT        (see previous DOT note)
   FD :HOR RT 90
   FD :VERT LT 90
   JUMP :HOR (:VERT + :GRAVITY) :GRAVITY
   END
```

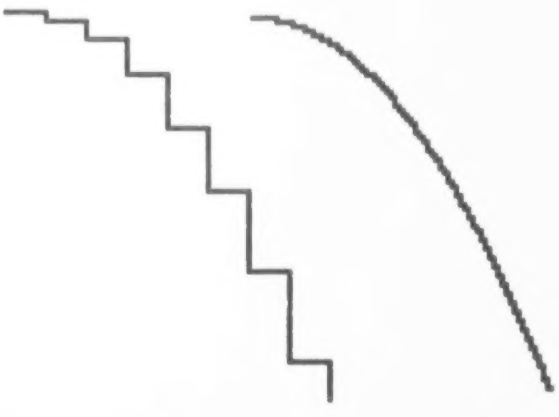
Note that turns of RT 90 and LT 90 are used to change from horizontal motion to vertical motion and back again.

```
TO SETUP
   HOME CS
   PU FD 100 RT 90 BK 100
   END
```

Type SETUP to position the turtle in the upper left part of the screen. Then select three numbers for JUMP and begin experimenting. JUMP 10 2 4 is a suggested start.



After some exploration, eliminate the DOT command line from your procedure, and try to find the numbers which will draw a smooth parabolic shape.



Don't get the idea that this is the only way to draw parabolas. If you already have read "Tipps for Teachers" in this issue, you may have wondered if parabolas could be produced using the scrunch commands. Why not give them a try?

FD 100!

Tom Lough teaches undergraduate physics at Piedmont Virginia Community College near Charlottesville, VA, and is the editor of The National Logo Exchange.

Logo Books from Prentice-Hall

Prentice-Hall of Australia has published two Logo books which will be extremely helpful to Logo teachers and students.

Learning Logo on the Apple II, by Anne McDougall, Tony Adams, and Pauline Adams, contains over 200 pages of Logo instructions, examples, and projects for both MIT (Terrapin and Krell) and LCSI (Apple) Logo versions. Unlike many other Logo books for beginners, one third of the text is devoted to the use of Logo beyond turtle graphics. Several helpful appendices, including one with a side by side listing of the commands of the MIT and LCSI Logo versions, round out this excellent book. It is available in the United States now for $19.95. French and Dutch translations are underway, as is an edition for the TRS-80 Color Logo by Radio Shack.

Let's Talk Apple Turtle, by Liddy Nevile and Carolyn Downing, is a 74 page activity book for children. It contains a rich mixture of both on and off computer projects, each selected to help children discover concepts underlying particular aspects of Logo work. Coloring, drawing, and paper folding are used to full advantage to amplify corresponding Logo activities. The teachers' edition is especially well designed. This book will be available in the United States shortly, with a list price anticipated to be under $10. The teachers' edition will have a list price most likely under $25. An edition for Commodore Logo will be available soon.

For more information, write Prentice-Hall, Inc. Englewood Cliffs, NJ 07632 USA, or Prentice-Hall of Australia, 209 Glenhuntly Road, Elsternwick, Victoria 3185 Australia.

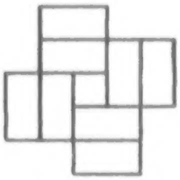
NLXual Challenges

with Robs Muir

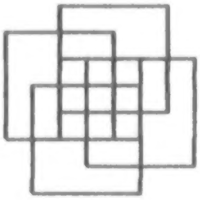
I make a point of offering a new Logo challenge to my students on a regular basis. Every week, I either bring in a hard copy printout of an interesting Logo design, or I sketch a problem on the chalkboard (sometimes "low tech" is faster and easier!). Many of my fourth and fifth grade students live for these problems, and can't wait to see if they will be among the first to crack the next one.

One Sunday evening, a few months ago, I realized that I didn't have a new challenge to bring into class on Monday. To compound the problem, I didn't have an inkling of what I would do - a sure sign of a relaxing weekend well spent! After cranking up the Apple and booting Logo, I found myself staring at a blank fullscreen editor. Since all procedures begin with TO and end with END, I knew that all I had to do was fill in the middle. Serendipity and hacking are old friends; they both came to visit that night.

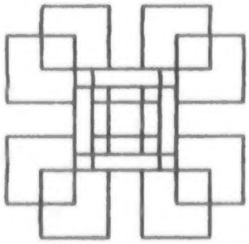
This month's NLXual Challenge is the outgrowth of that short hacking session. Simply stated, write the procedures necessary to recreate these murderous mysteries.



MURDER.MYSTERY1



MURDER.MYSTERY2



MURDER.MYSTERY3

"Mathematician George Polya has argued that general methods for solving problems should be taught ... For example, Polya recommends that whenever we approach a problem we should run through a mental checklist of heuristic questions such as: Can this problem be subdivided into simpler problems? Can this problem be related to a problem I already know how to solve? Turtle geometry lends itself to this exercise." (S. Papert, 1980)

As with all of the NLXual Challenges, if you want to receive a possible solution, send a copy of your solution (or your attempt) with a letter size self addressed stamped envelope to NLXual Challenges, Attn: Robs Muir, 1688 Denver Avenue, Claremont, CA 91711. We will compile contributions and send you the most elegant solution.

If you would like to submit a challenge or puzzle for this column, send your ideas to the above address.

Robs Muir is a teacher in the Claremont CA Unified School District and Claremont Graduate School. He presents Logo workshops in the southern California area.

The National Logo Exchange, copyright 1984 by Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly, September through May, $25 (US) per year, mailed FIRST CLASS from Charlottesville, Va. $5 (US) additional per year for addresses outside the USA, Canada, and Mexico. The opinions expressed by the authors are not necessarily those of The National Logo Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress Street, Salem, MA 01970. Address other correspondence to: The National Logo Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor..Tom Lough

ISSN 0734-1717

The National LOGO Exchange

Volume 2 Number 9 FORWARD 100! May 1984

Speaking of Logo, Now Tracy Can

by Kendall Young
and Glen Bull

Seymour Papert explores the conventional role of computers in education by asking,

> "What if there were only
> one pencil per classroom?"

One pencil in a classroom is better than none at all. But how would it change the classroom if every child had one? He then asks the same question about computers. What if every child in a classroom had a computer?

ONE CHILD, ONE COMPUTER

The effect of a personal computer can be even more striking if a child is disabled. Tracy Fewell of Crozet, VA, is 14 years old. He has spastic cerebral palsy and a speech disorder. In January, Tracy received an Apple computer with Logo. One day in class a month later, he presented his first oral report ever.

The muscles of Tracy's mouth are partially paralyzed. They are abnormally weak, slow, and uncoordinated. He can only produce five consonants. His longest phrase is four to five words in length. He can be understood only with difficulty, and only by those who have known him for a long period of time.

Tracy was able to deliver his first oral report because he had received a Votrax speech synthesizer for his computer, purchased with funds donated by a local Kiwanis Club.

The speech synthesizer is as easy to use as a printer, and is even less expensive. It attaches to the computer through a printer port. When the computer sends a line of the text to the printer port, the speech synthesizer speaks the sentence instead of printing it.

Photo by Jim Carpenter
Daily Progress, Charlottesville, Va.

LOGO AND SPEECH

A short Logo procedure allows Tracy to say anything he can type:

```
TO SAY :SENTENCE
   OUTDEV 1
   PRINT :SENTENCE
   OUTDEV 0
   END
```

Since his sixth grade classmates were learning Logo, the very first thing Tracy did was to type in instructions so they all could hear what their names sounded like with his new "voice."

For his first report, Tracy used the Logo editor to type in his homework. He used the synthesizer to present a social studies report the next day, with a procedure called TELL.STORY.

Before he had his computer, Tracy had to write each word laboriously by hand. He tended to write telegraphically, and leave out articles such as "the". Verbs such as "is" and "are" were used interchangeably. Spelling was irregular. Word endings such as "ed" were left off.

Speaking continued

Now he uses the TELL.STORY procedure to scan his work line by line. As he listens, he detects and corrects over half of all his errors without prompting. He has learned that things he "says" simply do not sound right when words such as "the" are left out.

PLEASE CALL ON ME

The import of Tracy's new-found oral capability is difficult to estimate. In previous years, Tracy has not been able to answer aloud in class. With a Logo program called QUICK.ANSWER, developed by educational specialist Cheryl Wissick, he can raise his hand and press a single key to select from a bank of answers for that day's work.

LOGO WHIZ

Tracy also has a learning disability. He is older than most of his classmates, because he has not advanced as quickly as they have in reading and spelling. With the combination of these physical and learning disabilities, it might be expected that Tracy would also lag behind in the computer curriculum.

Instead, other children have come to Tracy with their computer problems. He has been given the tag of "whiz kid" by his classmates. This has been good for Tracy's self-esteem. It also demonstrates the truth underlying Papert's observation about the advantages of giving every child unlimited access to computers.

Now that Tracy has his own "pencil", he can erase some of his educational barriers. Let us learn how to use Logo to do the same for others.

Note: For more information on the Votrax Personal Speech System, write to Votrax, 500 Stephenson Highway, Troy, MI 48084.

Kendall Young is a speech-language pathologist in the Albemarle County VA School System, and uses Logo regularly in her programs. Glen Bull is a professor in the University of Virginia School of Education's Department of Speech Pathology and Audiology, and is a regular columnist for The National Logo Exchange.

From The Editor

It is difficult to believe that another school year is drawing to a close. This time, the realization comes in a new way for us. Our son, Kyser, will celebrate his first birthday in May. It seems as if we brought him home from the hosiptal just a couple of weeks ago!

As I thought about Kyser's development, I began to see parallels with Logo. At first, everyone was excited at his arrival. He was the new kid on the block, "practically perfect in every way."

Logo also started off with the excitement of a new baby. Now that it has been around for a while, perhaps some of the excitement is wearing off. Articles are appearing which ask thoughtful penetrating questions. We are seeing that there are limitations and inappropriate uses. There are ineffective ways to teach Logo, and ineffective ways to teach with Logo.

Perhaps we are seeing a sort of maturing process. Preliminary research has not yielded all the glowing results expected. Yet, the potential and promise of Logo remain nearly undiminished. Why?

The use of Logo as a teaching tool in content areas beckons. Once students have some familiarity with the language, they can use it to study and learn other subjects.

Logo applications, such as with special populations, have begun to be reported. The need for such applications is critical. Logo can fill a vital part of the void.

Creative Logo people, such as you, our readers, continue to find new ways to use this language to help students extend their boundaries.

As we follow another year of Logo development, let us each do our part to take full advantage of this educational tool, for our students' sakes.

Best wishes for a rewarding summer! Don't forget to set aside time for reading, thinking, and writing. Here's hoping our paths cross.

FD 100! Tom

TIPPS for TEACHERS

by
Steve Tipps

How Now, Output

OUTPUT is a Logo command which people tend to avoid. Perhaps it frightens us after the simple English words (FORWARD, BACK) which introduce Logo.

OUTPUT is really much simpler than it may seem. Demystifying OUPUT hinges on getting it to do something meaningful for us.

WHICH COLOR IS WHAT NUMBER

I have a terrible time remembering whether blue or orange or green is 1 or 5 or 10. Switching from computer to computer and Logo version to Logo version compounds my confusion. I didn't want to look up the colors each time. What I really wanted was to set the pencolor with the name of the color.

Procedures with OUTPUT are good for remembering what numbers go with which colors. Each color procedure has one simple OUTPUT statement.

```
TO GREEN       TO ORANGE      TO BLUE
   OUTPUT 2       OUTPUT 4       OUTPUT 5
   END            END            END
```

For the Apple, pencolors and background colors could be set with the same color procedures (discounting the color interference problem.) Each procedure does only one thing - it outputs the number for that color. PENCOLOR uses that number, and I only have to type which color I want by name.

I can draw a green line or a blue square or an orange circle.

```
PENCOLOR GREEN FD 100          MIT
SETPC GREEN FD 100             LCSI

PENCOLOR BLUE SQUARE 100       MIT
SETPC BLUE SQUARE 100          LCSI

PENCOLOR ORANGE CIRCLE 5       MIT
SETPC ORANGE CIRCLE 5          LCSI
```

The numbers for colors change according to the version and machine, of course. You can also change the name of the color. PURPLE may be a better name than VIOLET or MAGENTA. IBM "brown" is actually a mustard yellow color. Make a file of color procedures for a tool kit.

ALL AHEAD GREEN

Since GREEN is a procedure which OUTPUTs the number 2, GREEN also can give the number 2 to other commands and procedures. That command or procedure can use the 2 the way it uses other numbers.

You can have the turtle move GREEN ahead.

```
FORWARD GREEN
or
REPEAT 100 [FORWARD ORANGE]
```

The turtle can "turn" BLUE or ORANGE.

```
RIGHT BLUE
or
REPEAT 360/ORANGE [FORWARD GREEN LEFT
ORANGE]
```

Colorful geometry yields to colorful arithmetic with new OUTPUT procedures for other colors. The values of other colors could be set according to your Logo, or arbitrarily, for number games.

```
TO YELLOW            TO RED
   OUTPUT ______        OUTPUT ______
   END                  END

TO BROWN
   OUTPUT ______
   END

TO COLOR.NUMBERS
   TEXTSCREEN CLEARTEXT  SETCURSOR [0 0]
      or  use NODRAW (MIT versions)
   PRINT [ YELLOW TIMES RED IS ...]
   PRINT YELLOW * RED
   PRINT [ RED TIMES BROWN IS ...]
   PRINT RED * BROWN
   PRINT [ BROWN TIMES YELLOW IS ...]
   PRINT BROWN * YELLOW
   PRINT [CAN YOU FIGURE OUT THE]
   PRINT [NUMBERS FOR THE COLORS?]
   END
```

When you have guessed what the color numbers are, you can check them by typing

```
PRINT YELLOW
PRINT RED
PRINT BROWN
```

Tipps continued

Other mystery numbers games could be made with procedures named for flowers or cars or stars.

A TRIVIAL QUIZ

Use OUTPUT in a procedure when you want it to tell another procedure or command something. The information could be a number or a word or a list. You can make a trivia game or quiz with OUTPUT (the short form, OP, can be used also).

```
TO TEXAS.CAPITAL?   TO FONZIE.ACTOR?
  OP "AUSTIN          OP [HENRY WINKLER]
  END                 END

TO SAREK.SON?       TO PRESIDENT.AFTER.NIXON?
  OUTPUT "SPOCK       OUTPUT [GERALD FORD]
  END                 END
```

POTS becomes a menu of questions to be chosen.

```
     POTS
     TO TEXAS.CAPITAL?
     TO FONZIE.ACTOR?
     TO SAREK.SON?
     TO PRESIDENT.AFTER.NIXON?
```

When an answer is guessed, you can check it by typing PRINT and the OUTPUT procedure.

```
     PRINT SAREK.SON?
     SPOCK
```

WHY INDEED?

You might ask, "Why OUTPUT instead of PRINT?" In the example, PRINT would do as well because we only need the information at the TOPLEVEL in Logo. However, when information is passed between procedures below TOPLEVEL, OUTPUT is required.

Convert the quiz to a procedure with questions and answers.

```
     TO QUIZ
        PRINT [ANSWER ONE OF THE FOLLOWING
QUESTIONS.]
        PRINT [1 WHO WAS SAREK'S SON?]
        PRINT [2 WHAT IS THE CAPITAL OF
TEXAS?]
        PRINT [3 WHO WAS PRESIDENT AFTER
NIXON?]
        PRINT [4 WHAT IS THE MOTTO OF THE
USA?]
        PRINT [5 WHAT ACTOR PLAYS FONZIE?]
        PRINT []
        WAIT 50
        PRINT [WHICH ANSWER DO YOU WANT?]
        PRINT [CHOOSE A NUMBER 1 THROUGH 5.]
        PRINT CHOOSE.ANSWER
        QUIZ
        END

     TO CHOOSE.ANSWER
        MAKE "NUMBER READCHAR
        IF :NUMBER = 1 [OUTPUT "SPOCK]
        IF :NUMBER = 2 [OUTPUT "AUSTIN]
        IF :NUMBER = 3 [OUTPUT [GERALD FORD]]
        IF :NUMBER = 4 [OUTPUT [E PLURIBUS
UNUM]]
        IF :NUMBER = 5 [OUTPUT [HENRY
WINKLER]]
        IF :NUMBER > 5 [OUTPUT [ PLEASE
CHOOSE A NUMBER 1 THROUGH 5.]]
        END
```

The answers are in CHOOSE.ANSWER. Depending on the question chosen, CHOOSE.ANSWER supplies the correct answer to PRINT in QUIZ. (You may want a more effective trap for numbers not 1 through 5.)

Compare how PRINT and OUTPUT were used with the instructions about choosing a number. The phrase was used immediately with PRINT, but was to be passed up a level from CHOOSE.ANSWER. This illustration also shows another feature of OUTPUT. As soon as OUTPUT is executed, the procedure automatically stops. If the number is 4, the procedure outputs [E PLURIBUS UNUM] and stops.

NOT SO TRIVIAL

A slightly fancier trivia quiz can also be made using a list of answers rather than the cumbersome CHOOSE.ANSWER procedure. This version is also more modular. Here are the questions.

```
     TO QUESTIONS
        TEXTSCREEN CLEARTEXT SETCURSOR [0 0]
        PRINT [ANSWER ONE OF THE FOLLOWING
QUESTIONS.]
        PRINT [1 WHO WAS SAREK'S SON?]
        PRINT [2 WHAT IS THE CAPITAL OF
TEXAS?]
        PRINT [3 WHO WAS PRESIDENT AFTER
NIXON?]
        PRINT [4 WHAT IS THE MOTTO OF THE
USA?]
        PRINT [5 WHAT ACTOR PLAYS FONZIE?]
        PRINT []
        END
```

Tipps continued

The answers are organized into a list.

```
     TO ANSWER.LIST
        MAKE  "ANSWERS [SPOCK AUSTIN  [GERALD
FORD][E PLURIBUS UNUM][HENRY WINKLER]]
        END
```

A separate module for answering has several parts. First, the cursor moves to a new location and asks you to pick a question.

```
     TO ANSWER
        SETCURSOR LIST 15 4
        PRINT [WHICH QUESTION DO YOU WANT  TO
ANSWER?]
        MAKE "NUMBER READCHAR
        PRINT CHOOSE.ANSWER2
        WAIT 20
        END
```

CHOOSE.ANSWER2 gives instructions and waits until you are ready to check your answer. Then it picks out the answer from the list which goes with the question you picked. This answer is picked with the command ITEM and OUTPUT to PRINT in ANSWER. If a number more than five is picked, additional instructions are given. (ITEM is a primitive in LCSI, Commodore , and new Terrapin 2.0. See <u>Logo for the Apple II</u>, Harold Abelson, McGraw-Hill, for information on writing ITEM for Terrapin 1.0 and Krell).

```
     TO CHOOSE.ANSWER2
        SETCURSOR [20 10]
        PRINT [TOUCH ANY KEY WHEN YOU WANT TO
CHECK YOUR ANSWER.]
        MAKE "GO.ON READCHAR
        IF :NUMBER > 5 [OUTPUT [PLEASE CHOOSE
A NUMBER 1 TO 5.]]
        OUTPUT ITEM :NUMBER :ANSWERS
        END
```

All the parts of the quiz are then organized into QUIZ.2, which is recursive.

```
     TO QUIZ.2
        ANSWER.LIST
        QUESTIONS
        ANSWER
        QUIZ.2
        END
```

A challenge would be to expand the questions and answer lists. You could also make the program more interactive by actually reading the answer from the keyboard. Try to eliminate the questions which have already been asked and answered. Then see if you can replace the used up questions with new questions. Soon you will have an electronic trivia game.

CLOSE ENCOUNTERS OF THE OUTPUT KIND

The quiz and game could be done with PRINT and without OUTPUT. However, they demonstrate how OUTPUT works in a tangible way. OUTPUT tells an outside procedure something an inside procedure knows. Sophisticated uses of OUTPUT capitalize on its ability to pass information between procedures in ways that PRINT cannot.

Creating procedures which do calculations is one example. Perhaps the total of the boys and girls and the percentage of boys and girls in a class are needed. You could write a procedure which would add the variable :BOYS to the variable :GIRLS for a TOTAL.

```
     TO TOTAL :BOYS :GIRLS
        OUTPUT :BOYS + :GIRLS
        END

     PRINT TOTAL 15 10
     25
```

This total procedure would be used to calculate percentage.

```
     TO PERCENTAGE :BOYS :GIRLS
        PRINT [THE PERCENT OF BOYS IS ...]
        PRINT  :BOYS  *  100  /  TOTAL  :BOYS
:GIRLS
        PRINT [THE PERCENT OF GIRLS IS ...]
        PRINT  :GIRLS  *  100 /  TOTAL  :BOYS
:GIRLS
        END

     PRINT PERCENTAGE 15 10
     THE PERCENT OF BOYS IS ...
     60
     THE PERCENT OF GIRLS IS ...
     40
```

Notice that the variables :BOYS and :GIRLS are input to PERCENTAGE and are given to TOTAL where they are added. Then the TOTAL is used in calculating the percent. In this case, OUTPUT is absolutely necessary. Try doing this with PRINT in TOTAL. You will see that PRINT will put the information on the screen, but will not let PERCENTAGE use it in calculations.

Tipps . . .

OUTPUT ANONYMOUS

You have already encountered several Logo commands which are natural outputters. HEADING outputs the current direction of the turtle, XCOR and YCOR output the current location of the turtle, BACKGROUND and PENCOLOR (LCSI) output the current background and pencolor, and TURTLESTATE (MIT) or DRAWSTATE (Commodore Logo) outputs a list of information about the current state of the turtle.

All of these natural outputters want to be told what to do with the information they are holding. Without PRINT (or some other command), the outputters are incomplete. The LCSI versions of Logo have a rather nasty complaint:

```
HEADING
I DON'T KNOW WHAT TO DO WITH 90
```

The MIT versions have a milder complaint:

```
XCOR
RESULT: 10
```

Both versions prefer to used the information in some way such as with PRINT.

```
TO WHERE?
   PRINT HEADING
   PRINT XCOR
   PRINT YCOR
   END
```

If you need to remember where the turtle is and which way it is pointing so that it can get back, the natural outputters are perfect.

```
TO REMEMBER.WHERE
   MAKE "START.DIRECTION HEADING
   MAKE "START.X XCOR
   MAKE "START.Y YCOR
   END
```

After moving around the screen, the turtle can get back to its starting place.

```
TO GO.BACK
   PU
   SETX :START.X
   SETY :START.Y
   SETHEADING :START.DIRECTION
   END
```

DON'T BE PUT OFF BY OUTPUT

OUTPUT is really not mysterious or frightening, although it can be tricky. When you are working on Logo this summer take a little time for experimenting with OUTPUT. The examples here only hint at some of the powerful things which OUTPUT allows you to do.

Being comfortable with with this handy little command will enable you to move into list processing and list structure which are at the heart of Logo.

Steve Tipps is a professor in the University of Virginia's School of Education, and presents Logo workshops for school systems throughout the United States.

NLXionary

A lectionary of selected Logo readings

compiled by Griff Wigley

Welcome to the NLXionary! Each month, we will present a lectionary of articles on Logo from other publications for your reading pleasure.

RECENT

inCider Magazine, March 84, "Logo - Where is the Pony?" by Molly Watt

>Watt describes the extreme positions of the Logo optimists and the Logo pessimists, validating and discounting some of the claims made by each. Her advice? Since Watt's title is a puzzle until you read the article, my answer will be cloudy as well: the story of Icarus from Greek mythology gives a hint - "In medio tutissimus ibis." Latin lives like Logo!

Antic, The Atari Resource Magazine, March 84, "Logo's Lineage, The Canadian Connection," by Ian Chadwick

>Great Material for a Logo trivia contest. Who studied with Papert under Piaget, had a computer company that went bankrupt, and was one of the founders of LCSI? What school in Canada was a site for Logo experimentation in the early seventies? When and where was LCSI formed? Who was the main technical genius behind LCSI's Logo development? Who provided the business skills? The article also updates Logo development within the Canadian educational system.

NLXionary continued

Home Computer Magazine, Volume 4, Number 1, 3 Logo articles

>Formerly 99'er Magazine, devoted entirely to Texas Instruments, HCM now covers Apple, Commodore, IBM, and still TI. Its regular Logo section, "Logo Times," sticks to Terrapin's version of Logo for the Apple and the Commodore 64, LCSI's Logo for the IBM PC and PCjr, and, of course, TI Logo. The three articles are project oriented, and provide a challenging glimpse of three different worlds within Logo.

>"What is Logo," by Robert Ackerman, describes Logo in the framework of a turtle graphics project building a city. "Lyrical Logo," by Logo veteran Henry Gorman, shows the list handling abilities of Logo in building a poetry generator. Gorman uses the sprites of TI and Commodore Logo in the third article, "Logo Shoots for the Moon."

Electronic Learning, March 84, "Word Processing in a Logo Environment," by Sharon Kramer

>A description of the author's method for teaching word processing to high school dropouts. Logo as a learning philosophy, not computer language, is the basis for her approach. I liked hearing how teaching Logo can influence a teacher's style in other subjects. Although I think we should be careful about calling any process-oriented approach to learning a "Logo approach", if the metaphor fits and can make dents in our product-oriented schools, let's use it.

ANCIENT GEM

Creative Computing, October 1982, "What is Logo?" by Molly Watt

>Not so much an article on what Logo is, but an account of the many processes that occur and roles that are played in a Logo class. It contains many hints about teaching Logo, and a novel listing of what students say they learn in a Logo class.

Griff Wigley is a co-owner of Family Computing, Inc., a facilitator in the Faribault (MN) Public Schools, and a school board member of Prairie Creek Community School.

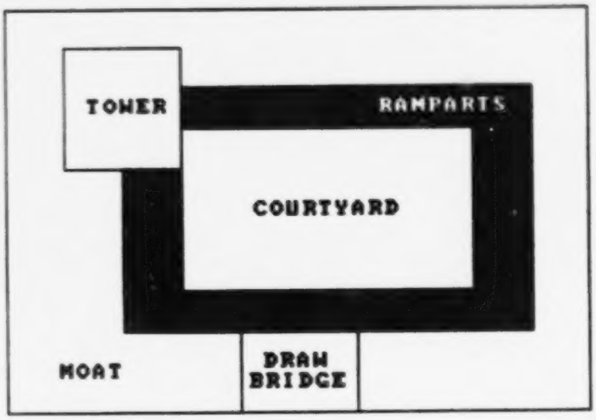
MICROWORLDS

by

Glen Bull

ETV: Educational TurtleVentures Part II

The turtle is exploring an enchanted castle - over the drawbridge, into the courtyard, and up a tower overlooking a moat.



Where is the turtle? Let's ask.

```
TELL WHERE?
TOWER
```

What is in the TOWER with us? We can ask that too.

```
TELL WHAT?
CANDLE KEY MIRROR
```

The CANDLE may be magic. (This is an enchanted castle.) But there are no matches. Wait. We may have noticed some on the RAMPARTS. The turtle moves from the TOWER to the wall overlooking the moat. We ask where the turtle is.

```
TELL WHERE?
RAMPARTS
```

```
TELL WHAT?
MATCHES SWORD BUCKET
```

We can take the MATCHES back to the tower to light the CANDLE. We tell the turtle to pick up the matches.

```
CARRY MATCHES
```

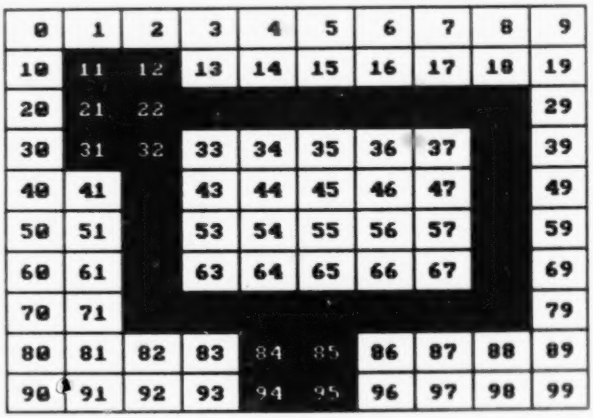
This type of adventure can be written in Logo.

Microworlds continued

LOOKING BACK

Last month, the screen was divided into a 10 by 10 grid of 100 blocks, labeled 0 through 99 with a procedure called SETUP, and made into the floorplan of a home. The numbers for the width and height of your screen may be different from 320 and 250.

```
     TO SETUP
        MAKE "SCREEN.WIDTH 320
        MAKE "SCREEN.HEIGHT 250
        MAKE "NO.OF.ROWS 10
        MAKE "NO.OF.COLS 10
        MAKE  "SQUARE.WIDTH  :SCREEN.WIDTH  /
:NO.OF.COLS
        MAKE "SQUARE.HEIGHT :SCREEN.HEIGHT  /
:NO.OF.ROWS
        MAKE  "VERTICAL.OFFSET :SCREEN.HEIGHT
/ 2 - 1
        MAKE "HORIZONTAL.OFFSET :SCREEN.WIDTH
/ 2 - 1
        END
```



0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22							29
30	31	32	33	34	35	36	37		39
40	41		43	44	45	46	47		49
50	51		53	54	55	56	57		59
60	61		63	64	65	66	67		69
70	71								79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

The procedure BLOCK? was created to find out what block the turtle was sitting on .

```
    TO BLOCK?
       LOCAL "X.Y
       MAKE "X.Y GRID.LOCATION
       OUTPUT WORD LAST :X.Y FIRST :X.Y
       END

   TO GRID.LOCATION
     LOCAL "X LOCAL "Y
     MAKE "X INT (XCOR + :HORIZONTAL.OFFSET)
/ :SQUARE.WIDTH
     MAKE "Y INT (YCOR + :VERTICAL.OFFSET) /
:SQUARE.HEIGHT
     MAKE "Y 9 - :Y
     OUTPUT LIST :X :Y
     END
```

When the turtle is sitting on block 37, the result looks like this:

```
     PRINT BLOCK?
     37
```

WHERE IS THE TURTLE?

Notice that the castle TOWER sits on these six blocks:

11 12 21 22 31 32

If we make a list of the blocks occupied by each place, we can find out where the turtle is.

The procedure NAME.PLACES names the blocks which make up the various parts of the castle. You must add the statements for the blocks in the COURTYARD, the RAMPARTS, and the MOAT before this procedure is complete.

```
  TO NAME.PLACES
     MAKE "TOWER.SPACE [11 12 21 22 31 32]
     MAKE "DRAWBRIDGE.SPACE [84 85 94 95]
        .
        .
        .
     END
```

Next, we can find out where the turtle is by asking if the block the turtle is on is a member of that place. MEMBERP is the command which allows us to ask this question. You must run the completed NAME.PLACES procedure before you can use the WHERE? procedure.

```
  TO WHERE?
     LOCAL "LOCATION
     MAKE "LOCATION BLOCK?
     IF  MEMBERP :LOCATION  :TOWER.SPACE  [OP
"TOWER]
     IF  MEMBERP :LOCATION  :DRAWBRIDGE.SPACE
[OP "DRAWBRIDGE]
     IF  MEMBERP  :LOCATION  :MOAT.SPACE  [OP
"MOAT]
     IF  MEMBERP  :LOCATION  :COURTYARD.SPACE
[OP "COURTYARD]
     IF MEMBERP :LOCATION :RAMPARTS.SPACE [OP
"RAMPARTS]
     OP [LOCATION NOT IDENTIFIED]
     END
```

The procedure works like this:

```
     PRINT WHERE?
     COURTYARD
```

Personally, I prefer TELL WHERE? in place of PRINT WHERE? as an instruction.

More Microworlds

Here is a Logo procedure which allows us to use TELL instead of PRINT:

```
TO TELL :INPUT
   PRINT :INPUT
END
```

WHAT IS THERE?

Now that we can find out where the turtle is, we can make a list of the things in each place. Here is how.

```
TO NAME.THINGS
   MAKE "TOWER [CANDLE KEY MIRROR]
   MAKE "DRAWBRIDGE [HORSESHOE ROPE]
   MAKE "MOAT [BOOT CAN SHIELD]
   MAKE "COURTYARD [MOP CHEST GRANOLA]
   MAKE "RAMPARTS [MATCHES SWORD BUCKET]
   MAKE "TURTLE []
   END
```

You can place anything you want in the brackets after the name of each place. In the last line I have added a list of the things the turtle is carrying. Right now, the turtle is not carrying anything, so nothing is between the brackets after the word TURTLE.

After describing the things in each place, the procedure must be run by typing NAME.THINGS. After doing that, we can find out what those things are by entering:

```
PRINT THING "TOWER
CANDLE KEY MIRROR
```

If we want to know what is in the place where the turtle is, the word WHERE? can be substituted for the name of the place.

```
TO WHAT?
   OUTPUT THING WHERE?
   END
```

With this procedure, we can ask what is in the place where the turtle is. If the turtle is on the DRAWBRIDGE we would find:

```
TELL WHAT?
HORSESHOE ROPE
```

TURTLE PICKUP

The turtle needs a way to carry things from one place to another. If the turtle picks up a box of matches on the RAMPARTS and carries it to the TOWER, we should no longer see matches in the list of things on the RAMPARTS. Two steps are necessary to accomplish this:

(1) We must be able to add to the list of things the turtle is carrying.
(2) We must be able to delete from the list of things attached to a place.

Adding things to a list is straightforward. To add an item to a list, LPUT can be used to put it into the list as the last item.

```
TO ADD :ITEM :LIST
   MAKE :LIST LPUT :ITEM THING :LIST
   END
```

Let's use this procedure to add a flute to the list of items on the DRAWBRIDGE.

```
TELL WHERE?
DRAWBRIDGE

TELL WHAT?
HORSESHOE ROPE

ADD "FLUTE "DRAWBRIDGE
TELL WHAT?
HORSESHOE ROPE FLUTE
```

Deleting an item from a list is a little more tricky. It would be nice if this were a built-in command. Since it is not, you may find this procedure useful for other Logo programs.

```
TO DELETE :ITEM :LIST
  MAKE :LIST DELETE.LOOP :ITEM THING :LIST []
  END

TO DELETE.LOOP :ITEM :OLD.LIST :NEW.LIST
  IF  EMPTYP  :OLD.LIST [PRINT SE :ITEM  [NOT
FOUND] OUTPUT :NEW.LIST]
  TEST :ITEM = FIRST :OLD.LIST
  IFTRUE  [OUTPUT SE  :NEW.LIST  BUTFIRST
:OLD.LIST]
  OP  DELETE.LOOP  :ITEM  BF  :OLD.LIST  SE
:NEW.LIST FIRST :OLD.LIST
  END
```

Let's use the DELETE command to remove ROPE from the list of things on the DRAWBRIDGE.

```
TELL WHAT?
HORSESHOE ROPE FLUTE

DELETE "ROPE "DRAWBRIDGE
TELL WHAT?
HORSESHOE FLUTE
```

Microworlds concluded

When the turtle selects an item to carry, we must add the item to the list of things the turtle is carrying, and delete it from the list of things where the turtle is. Before doing this, the procedure CARRY checks to make sure the item is present.

```
TO CARRY :ITEM
  TEST MEMBERP :ITEM WHAT?
  IFFALSE [PRINT SE :ITEM [NOT FOUND] STOP]
  ADD :ITEM "TURTLE
  DELETE :ITEM WHERE?
  END
```

TURTLE DROPOFF

The procedure DROP reverses this process. It removes the item from the list of things the turtle is carrying, and adds them to the place where the turtle is.

```
TO DROP :ITEM
  TEST MEMBERP :ITEM :TURTLE
  IFFALSE [PRINT SE :ITEM [NOT CARRIED]
STOP]
  ADD :ITEM WHERE?
  DELETE :ITEM "TURTLE
  END
```

Here is how the the turtle would carry the KEY from the TOWER to the CHEST in the COURTYARD. The turtle must begin in the TOWER.

```
TELL WHERE?
TOWER

TELL WHAT?
CANDLE KEY MIRROR

CARRY "KEY
TELL WHAT?
CANDLE MIRROR
```

After the turtle carries the KEY, it will no longer be found in the list of items in the TOWER. Now we can use turtle commands to move the turtle to the COURTYARD.

```
TELL WHERE?
COURTYARD

TELL WHAT?
MOP CHEST GRANOLA

DROP "KEY
TELL WHAT?
MOP CHEST GRANOLA KEY
```

At this point we might like to tell the turtle to unlock the chest, or to find out what else the turtle is carrying. Alas, that must be a summer project.

GETTING INTO LISTS

Teachers find that the transition from turtle graphics to the list processing commands of Logo is not always smooth. The turtle is concrete and visible, while lists are more abstract.

I believe that, as we experiment with list processing commands in the classroom, natural ways of introducing lists will be identified. This adventure project was developed to suggest some ways that lists can be employed.

You could have a timed scavenger hunt to see who could use the turtle to collect specified items the fastest. In a spelling word hunt, the turtle could be used to collect the week's spelling list from different parts of the castle. In a message hunt, a message could be formed from correctly spelled words scattered among incorrectly spelled words.

In language arts, the items could be related to categories. For example, the instructions could be "Collect all the verbs" (or nouns or adjectives). Products of Argentina could be collected in a social studies lesson.

***In the March issue, I included the following line:

```
MAKE "NUMBER COUNT FIRST :LIST + :NUMBER
```

The line should be:

```
MAKE "NUMBER (COUNT FIRST :LIST) + :NUMBER
```

Have a good summer. See you next fall.

Glen Bull is a professor is the University of Virginia's School of Education, and teaches Logo courses at both the graduate and undergraduate level.

Oops

The review of Sprite Logo in the April NLX incorrectly referred to the product as "Apple Sprite Logo." The correct name is "Sprite Logo for the Apple II Family." We apologize for any misconceptions this may have caused.

Give Me a Sine: Lissajou Logo

by
Tom Lough

Come with me on the second leg of an odyssey I began last year when a member of the audience at a Logo demonstration I was giving asked, "Can the turtle draw a sine wave?" I wrote an admittedly incomplete article for the January 1983 NLX which examined this question in some detail.

After working with parabolas and the acceleration which produces them (see "ParaboLogo," NLX April 1984), the sine question came back to me. I wondered whether the acceleration idea could be used to produce a sine wave.

IMAGINE

Imagine a mass connected to a spring hooked to a firm support. Eliminate all gravity and friction. When the mass is moved away from the support, the spring will pull it back. The force (or acceleration) due to the spring is directly proportional to the distance from the mass to the support.

When the mass has moved only a short distance from the support, the restoring force is small. When the mass is ten times as far away, the restoring force is ten times as large. (See information on Hooke's Law for more details.)

Set the mass in motion from rest. The force of the spring increases, slowing the mass until it stops momentarily and reverses its direction. Can you imagine the force of the spring now causing the mass to speed up as it goes back toward the support?

By the time the mass reaches the support, its speed has increased to the original value. After it moves past the support, the mass feels the force of the spring pulling it backwards, slowing it down. The force builds; the mass comes to a halt and reverses its direction once more.

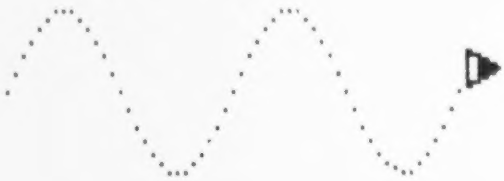
This back and forth pattern continues in a regular manner. If the mass moves sideways at a constant speed as it oscillates, its path is a sine wave!

WRITE A PROCEDURE

Once I understood how this all worked, the next step was to try to write a procedure to simulate the motion. I added a self imposed limitation - to use only FD, RT, and LT. After a number of attempts, I constructed the WAVE procedure listed at the end of the article.

I moved the turtle to the left side of the screen, set its heading to 90, and entered

```
WAVE 2 -5 .05 -5
```



The first input number sets the horizontal speed. The last three inputs are related to the vertical movement.

In the example, -5 is used to start the mass on an upward path. It represents the initial speed of the mass.

The number .05 represents the stiffness of the spring divided by the mass of the body. It is a constant which can be used to control the frequency of the motion. A stiff spring will not permit as wide a motion as one which is easy to stretch. Likewise, a large mass will not accelerate as quickly as a smaller one subjected to the same restoring force.

The last input number represents the distance of the mass from the support after each cycle of recursion. Since distance equals speed times time, we would expect this distance to be the initial speed times one (1) at the end of the first cycle. Thus, this input number is the same as that of the initial speed when the procedure is used to draw a sine wave.

WHAT ABOUT COSINE?

Since the cosine has the same shape as that of the sine wave, the WAVE procedure should be able to draw it also. We need to realize that the cosine begins at a different point than the sine, however.

Lissajou continued

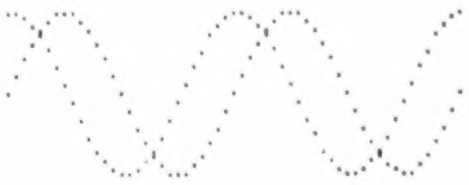
Imagine pulling the mass out from the support a certain distance and releasing it. Its motion will describe the cosine path.

Just as the mass is released, its speed is zero. The restoring force is at a maximum value, since the mass is at its greatest distance from the support.

If we want to draw a cosine wave the same size as the sine wave drawn earlier with WAVE, what displacement do we use for the last input? You might be tempted to say -5. Try it and see what happens.

What is needed here is the approximate maximum displacement for a wave which will have a maximum speed of -5 as it goes through the center. But how far is that distance?

Physics provides the answer. The maximum displacement is the maximum velocity divided by the square root of the spring/mass number. Thus, -5/SQRT(.05) could be used for the last input to WAVE to produce the desired cosine trace.



```
sine:     WAVE  2  -5  .05  -5
cosine:   WAVE  2   0  .05  -5/SQRT (.05)
```

Note the input value of zero for the initial speed for the cosine wave, and the term -5/SQRT (.05) as the final input.

Since the WAVE procedure uses only FD, RT, and LT, it is self referencing, and can be used to draw figures in any orientation.

LET'S LISSAJOU

Particularly important are the so-called "Lissajou figures." (See Frequency Measurement in Van Nostrand's Scientific Encyclopedia, for example.)

Lissajou figures are drawn in the same way you draw with an "Etch-a-Sketch" type of toy. You turn one knob for up and down pen movements, and another for sideways movements. The two controls operate at an angle of 90 degrees to each other.

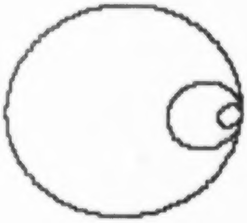
If you have ever tried to draw a circle with this type of toy, you have probably given up! It is very difficult, because ... you have to do a sine like motion on one knob while doing a cosine like motion on the other!

Aha! That means that we might be able to draw circles with our sine and cosine. The circle is a member of the Lissajou family. In fact, the entire family of figures can be drawn with sine and cosine in this manner.

The procedure FIGURE, listed at the end of the article, contains two WAVE procedures at 90 degrees to each other, and can draw Lissajou figures. When drawing them, I put the pen down, and use small inputs.

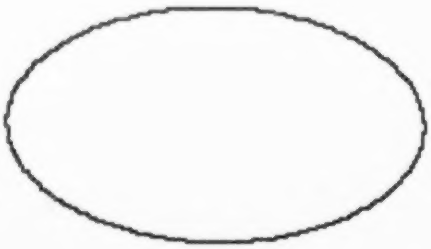
To draw a circle, the amplitudes of the sine and the cosine parts should be equal. For example, try

```
WAVE 1  .05  1  0  .05  1/SQRT (.05)
WAVE 1  .005  1  0  .005  1/SQRT (.005)
WAVE 1  .0005  1  0  .0005  1/SQRT (.0005)
```



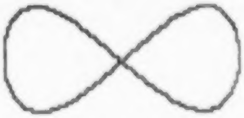
What would happen if the amplitudes were different? The result is an ellipse, another member of the Lissajou family. (By the way, this ellipse doesn't depend on the use of SCRUNCH or the location of a focus!)

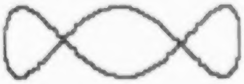
```
WAVE 1  .0005   1   0  .0005   2/SQRT (.0005)
```

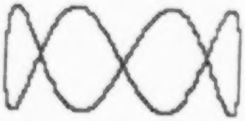


Lissajou continued

What happens if we change the frequency of one wave relative to another by using different spring/mass numbers? We discover the incredible richness of the Lissajou. Many other forms are possible. But, don't let me spoil your fun. Discover them for yourself!







As an interesting comparison, R. W. Lawler's article, "Designing Computer Based Microworlds," in the August 1982 issue of Byte magazine, lists a Lissajou procedure on page 158, which makes direct use of the SIN and COS commands, in addition to many other Logo primitives. Photographs of several Lissajou figures appear on page 160.

I hope that you and your students will be able to make use of the ideas presented here. They seem to be suited best for the physical sciences. If you apply them or extend them, please share your work with me. Thank you in advance. FD 100!

```
TO WAVE :HOR :VERT :K :DISPLACEMENT
     PD FD 1 BK 1 PU
     (or DOT POS or PD DOT POS PU)
     FD :HOR RT 90 FD :VERT LT 90
     WAVE  :HOR  (:VERT-:K*:DISPLACEMENT)  :K
(:DISPLACEMENT+:VERT-:K*:DISPLACEMENT)
     END

TO FIGURE :X :KX :XTOTAL :Y :KY :YTOTAL
     PD FD 1 BK 1 PU
     (or DOT POS  or PD DOT POS PU)
     FD :X RT 90 FD :Y LT 90
     FIGURE (:X-:KX*:XTOTAL) :KX (:XTOTAL+:X-
:KX*:XTOTAL)  (:Y-:KY*:YTOTAL)  :KY  (:YTOTAL
+:Y-:KY*:YTOTAL)
     END
```

Tom Lough uses Logo in teaching undergraduate physics classes at Piedmont Virginia Community College near Charlottesville, VA, and is the editor of The National Logo Exchange.

Teacher, Kids, and Logo

A review by Sue Mangiapene

Teacher, Kids, and Logo, written by Carolyn Green and Christi Jaeger, is one of the most comprehensive practical guides to Logo instruction I have seen. It is well written and well paced. The activities have been teacher tested. Teacher, Kids, and Logo shows teachers how to encourage children to draw on their creativity and gives them the opportunity to define their ideas in visual terms using the turtle's language.

Teacher, Kids, and Logo dedicates a separate chapter to each grade, K-6, and specifies techniques which may be used at each level. Each chapter contains objectives for the mathematical concepts, computer awareness, and programming concepts to be covered at that level. A step by step explanation follows to help teachers achieve the stated objectives with their children. This approach easily lends itself to individualized pacing.

The book contains advice on classroom demonstrations, provides concrete activities, and offers supplemental worksheets to help students tranfer their programming skills. Many of the ideas help children view the computer as a tool to be used in the real world, and help them relate their experiences to other computer applications.

Teacher, Kids, and Logo includes commands for both MIT and Apple Logo and can easily be adapted to other variations of the language. It also includes appendices which offer teaching aids and practice exercises. Many of these pages present wonderful ideas for further development of the logical and structural ideas needed in any programming experience.

I recommend Teacher, Kids, and Logo as an excellent resource for teachers who are eager to enhance their computer program.

EduComp Publications, 14242 Wyeth Avenue, Irvine, CA, 92714. Spiral bound 8" x 11", illustrated, 160 pages. $15.95 plus $2.00 shipping (CA residents add 6% sales tax)

Sue Mangiapane is the owner of Computers with Class, an educational consulting firm in Anaheim, CA, and is a research consultant for the California State University at Fullerton.

The National Logo Exchange
1983-84 Index Volume 2

Volume 2 Number 1 September 1983

Here Comes Atari Logo, by Shoneen Gervich
Tipps for Teachers, by Steve Tipps
    Borders and Frames
Microworlds, by Glen Bull,
    The Utilitarian Turtle
Slow Turtle Moves Clearly, by Tom Lough

Volume 2 Number 2 October 1983

Logo Mess Ups: Process vs. Product,
    by Nancy Parks Sopp
Tipps for Teachers, by Steve Tipps,
    The Truth about Numbers
Microworlds, by Glen Bull, Numbers and Lists
From Four Bugs to a Society of Turtles,
    by Michael Burke

Volume 2 Number 3 November 1983

Logo Stepping Stones for Young Minds,
    by Jan Cardwell
Tipps for Teachers, by Steve Tipps,
    Both Sides Now
Microworlds, by Glen Bull, Logo Grade Book,
    Part I: Grade Averager
Commodore Logo: Let's Take a Look,
    by Steve Tipps
NLXual Challenges by Robs Muir

Volume 2 Number 4 December 1983

Students Write the Book on Logo,
    by Lynne Mass
Pushing the Turtle, guest editorial
    by Steve Tipps
Problem Spaces in a Project Oriented Logo
    Environment, Part I, by Glen Bull and
    Steve Tipps
NLX ABC's Lead to List Processing Adventures,
    by Jim Fry
Tipps for Teachers, by Steve Tipps,
    The Issue of Instant
Microworlds, by Glen Bull, Logo Grade Book,
    Part II: Report Generator
NOVA's "Talking Turtle," A Compelling
    Documentary, a review by Andrew David
NLXual Challenges, by Robs Muir

Volume 2 Number 5 January 1984

The Hazards of Hacking, by Sharon Burrowes
Problem Spaces in a Project Oriented Logo
    Environment, Part II, by Glen Bull and
    Steve Tipps
Tipps for Teachers, by Steve Tipps,
    Triangle Thinking :SIDE1
Microworlds, by Glen Bull, Adventure Stories
    and English Skills
Of Sushi and Mushi, by Duane Yee
NLXual Challenges, by Robs Muir

Volume 2 Number 6 February 1984

Concrete Turtling, by Joan Rainey
As Easy As 1,2,3, a review by Pat Stevens
Tipps for Teachers, by Steve Tipps,
    Triangle Thinking :SIDE2
Microworlds, by Glen Bull, Logo Characters
Right Here in Turtle City, by Steve Tipps
NLXual Challenges, by Robs Muir

Volume 2 Number 7 March 1984

The Turtle Goes to College, by Lois Tepper
NLXionary of Logo Readings, by Griff Wigley
Papert Addresses Boston Logo Users' Group,
    by Nola Sheffer
Tipps for Teachers, by Steve Tipps,
    Triangle Thinking :SIDE3
Allen's Book is Something to Think About,
    a review by Glen Bull
Microworlds, by Glen Bull, Centering Titles
Sweet P is a Sweetheart, by Steve Tipps
NLXual Challenges, by Robs Muir

Volume 2 Number 8 April 1984

Breaking Up Is Hard To Do, by Cathy Frank
Tipps for Teachers; by Steve Tipps,
    Unsquare Square: The Story of Scrunch
Logo for the Sony, by Hillel Weintraub
Apple Sprite Logo, a review by Griff Wigley
Boston Logo Users' Group News: diSessa and
    the Dynaturtle, by Nola Sheffer
Microworlds, by Glen Bull, ETV: Educational
    TurtleVentures Part I
NLXionary of Logo Readings, by Griff Wigley
ParaboLogo, by Tom Lough
NLXual Challenges, by Robs Muir

Volume 2 Number 9 May 1984

Speaking of Logo, Now Tracy Can,
    by Kendall Young and Glen Bull
Tipps for Teachers, by Steve Tipps,
    How Now, Output
NLXionary of Logo Readings, by Griff Wigley
Microworlds, by Glen Bull, ETV: Educational
    TurtleVentures Part II
Give Me a Sine: Lissajou Logo, by Tom Lough
Teacher, Kids, and Logo,
    a review by Sue Mangiapane
NLXual Challenges, by Robs Muir

NLX Notices

The NLX announces its Dissertation and Thesis Repository. This collection of publications is designed to assist researchers in identifying related Logo work, and will serve as an adjunct to the NLX Logo Research Directory. We are inviting the submission of all dissertations and theses which involve Logo. Please send complete copies if possible. Include abstracts and any identification numbers (such as a university number or the Dissertation Abstracts index number). If complete copies are not available, please send as much information as possible. Mail your documents to NLX Dissertation and Thesis Repository, Attn: Barbara Elias, PO Box 5341, Charlottesville, VA 22905.

How can one find out what is being done with Logo and special populations in schools? Glen Bull and Ralph Stoudt discussed the matter with Dr. Sylvia Weir of MIT's Laboratory for Computer Science. Dr. Weir suggested that a case registry might be a worthwhile mechanism for people using Logo with special populations to get in touch with one another.

They are adopting Dr. Weir's suggestion to establish such a case registry. If you are using Logo with a special population (including communication disorders, learning disabilities, cognitive impairment, sensory impairment, and disorders involving physical handicaps such as cerebral palsy), or know of someone who is, please send information about it. Glen and Ralph will abstract each submission, catalog and index the entries, and set up a mechanism for sharing information.

They are especially interested in informal accounts from teachers and clinicians, but would also like to include any controlled research reported. Case registry information should include name, institution, address, phone number, type of special population, ways Logo is being used, results, and thoughts for future actions.

Send your reports to:

Logo Case Registry for Special Populations
Speech and Hearing Center
109 New Cabell Hall
University of Virginia
Charlottesville, Virginia 22903

Thank you for sending in your blue cards! We were pleased to hear from so many of our subscribers. We especially appreciated the ideas and suggestions you sent for upgrading the NLX. As a result, starting in September, we will be adding several new features which we hope will aid you in your Logo teaching. We would like to invite your participation in them now.

The first is a column of Logo classroom activities and tips for elementary teachers. If you would like to contribute ideas or share activities which have worked in your classroom, send a brief description to Donna Lanyi and Jane Toth, 2620 Taylor, Wooster, OH 44691.

We are also planning to begin an answer column for readers with questions about Logo programming techniques and the thinking leading to programming. If you have ever written a procedure that "should have worked" but didn't, here is your chance to find out why. Jim McCauley will answer several questions in his column each month. If you have a question you would like to be considered, send it to NLX Q&A, PO Box 5341, Charlottesville, VA 22905. Be sure to include a full description of the situation which led to the question, and a listing of the procedure(s).

NLXual Challenges

with Robs Muir

Of Squares and Strings

Certainly the most well known feature of the Logo language is the turtle. This computational object has features which we can recognize as having real world parallels. As a purposefully anthropomorphoric entity, the Logo turtle provides a powerful bridge between concrete objects and conceptual frameworks.

A turtle allows children to easily manipulate real world properties, i. e., distance and direction, to assist them in building mental models of the world.

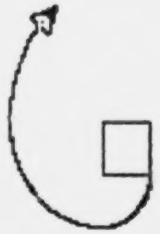
This month's problem, contributed by Tom Lough, starts with a real world situation, in which our goal will be to create a simulation of the concrete, using our computer monitor.

Challenges continued

Have you ever seen a child with a length of rope and a tree? At some point in play, the child will usually tie the rope to the trunk of the tree, grasp the other end, and begin walking around the tree. As the rope winds or unwinds, it causes the child to walk in a definite path. Let's use the turtle to explore this path.

Imagine a square prism standing vertically upon a sheet of paper. Wrapped around this square dowel is a length of string, with a pencil attached. If the string is kept taut, what pattern will the pencil inscribe on the paper as it revolves around the dowel, unwinding the string? Clearly, the general shape will be that of an increasingly widening spiral.

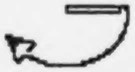
The first task is to write a program which will illustrate the path that the pencil will take as it unwinds from the square prism. You might begin with a small square, say, 20 by 20 turtle steps, in the center of the screen. Starting at the lower right corner of this square, have the turtle begin unwinding in a clockwise direction. Remember that our string must remain taut.

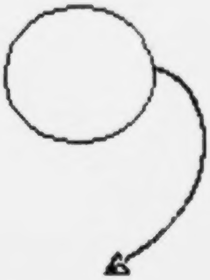


Notice the outwardly spiraling shape. Will the shape remain the same if, instead of a square, we used a triangular center of 20 turtle steps on each side? How might it differ?

Your complete challenge is to write a program of procedures that will draw the spiral created by centers of differing dimensions. For example, what would the spiral look like if the central core around which the string is attached was a rectangular prism of 3 by 20 turtle steps?

You may want to design an even more general solution, in which other shapes, such as hexagons, could be used at the center. In that case, examine the difference between the spiral you produced with the square, and one produced with a "circle."





As with all NLXual Challenges, if you would like to receive a possible solution, send me a copy of some of your work, along with a letter size self addressed stamped envelope. I'll send you a sample or two of the most interesting solutions submitted.

If you think of an interesting idea over the summer which could be developed into an NLXual Challenge, send me a brief description of it. See you in September.

NLXual Challenges, Attn: Robs Muir, 1688 Denver Avenue, Claremont, CA 91711.

Robs Muir is a teacher in the Claremont CA Unified School District and Claremont Graduate School. He presents Logo workshops in the southern California area.

The National Logo Exchange, copyright 1984 by Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly, September through May, $25 (US) per year, mailed FIRST CLASS from Charlottesville, Va. $5 (US) additional per year for addresses outside the USA, Canada, and Mexico. The opinions expressed by the authors are not necessarily those of The National Logo Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress Street, Salem, MA 01970. Address other correspondence to: The National Logo Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor.......................................Tom Lough

THE NATIONAL LOGO EXCHANGE

Volume 3 Number 1 FORWARD 100! September 1984

LOGO and DOS:

Classroom Partners

by

Nancy Parks Sopp

"Mrs. Sopp, can I initialize a new disk without losing my Logo procedure?" Bridgette asked.

"I'd like to use both disk drives with Logo," a colleague said. "Is there a way to do this?"

Andrew wanted to know, "Can I put my Koala Pad picture into a Logo procedure?"

I answered "Yes." to all three questions and taught my co-learners to use MIT Logo's DOS primitive on our Apple II computers.

DOS REVEALED

The disk operating system (DOS) consists of a group of commands for using the disk drive. LOAD, SAVE, DELETE, CATALOG, and BLOAD are examples. DOS commands are used usually with BASIC, but they are not part of the BASIC language. DOS commands are stored in the HELLO program that is on each initialized files disk.

Many DOS commands have direct Logo equivalents. READ is MIT Logo's equivalent for DOS's LOAD. ERASEFILE and ERASEPICT do the same job as DOS's DELETE.

Some DOS commands have no Logo equivalent. These can be used from within Logo, with the DOS primitive. Terrapin recently released their Version 2.0 of Logo. It does not support all the DOS commands that the older versions do. The remarks which follow are for older MIT Logo versions. Changes for Terrapin's 2.0 are noted at the end of the article.

The MIT Logo technical manual lists these DOS commands for use with the DOS primitive: DELETE, VERIFY, CATALOG, LOCK, UNLOCK, MON, NOMON, RENAME, BLOAD, BRUN, and BSAVE. Not listed, but supported, are INIT and D2 (when used after CATALOG or INIT HELLO).

HOW TO DOS

To use DOS commands, type DOS and the commands enclosed with square brackets []. For example, I showed Bridgette how to put her new disk in drive 1 and type

```
DOS [INIT HELLO]
```

The drive initialized her disk, with Logo and her workspace remaining intact. If Bridgett's disk were in drive 2, she could have typed

```
DOS [INIT HELLO, D2]
```

My colleague's desire to use drive 2 was satisfied similarly. He typed

```
DOS [CATALOG, D2]
```

Once he accessed drive 2, all further disk operations took place there until he typed

```
DOS [CATALOG, D1]
```

Now let's get to Andrew's question and the most powerful use my students and I have found for using DOS within Logo: BLOADing non-Logo pictures.

My students use the Koala Pad and Power Pad graphics tools to create pleasing, colorful pictures. They use Fontrix software to print words, slogans, and stories in stylized character sets. Images created with these and other graphics programs can be incorporated into Logo procedures with the DOS primitive.

Andrew wanted to use a scene he had created with his Koala Pad as a backdrop for Logo shapes. First, we tried to pull a fast one on Logo by adding .PICT to the end of his Koala picture name. We booted Logo and used READPICT to retrieve it. The screen remained blank and the computer froze up. Later, we found out why our trick didn't work.

There are two high resolution graphics screens in the Apple computer. These correspond to discrete memory locations within the

DOS continued

computer. Logo uses screen 1; the Koala Pad and Power Pad software use screen 2. The instructions in Andrew's Koala picture file automatically specified screen 2 as the target memory location. Logo was trying to load it into the memory location for screen 1. No wonder the machine was confused! We couldn't even crash Logo with control reset, and had to turn the computer off and reboot. We then learned to use Logo's DOS command, and typed

```
DOS [BLOAD PICTR.ISLAND, A$2000]
```

BLOAD is the DOS equivalent of READPICT. When Andrew saved his Koala picture ISLAND, the Koala Pad software added PICTR. to the front the filename. Thus, PICTR.ISLAND is the filename we needed to call with the DOS primitive. The A$2000 specifies the starting memory address for screen 1.

Andrew then used SAVEPICT to resave his picture from Logo. For reasons we do not understand, Andrew could not draw on the BLOADed Koala picture until he had used SAVEPICT, then READPICT.

He followed the same steps later with a picture made with the Power Pad.

GETTING FANCY

Joe made a poster with Fontrix and Logo. He first used Fontrix to print the name of this favorite rock group on the screen. He saved the image, then booted up Logo and retrieved the image with

```
DOS [BLOAD MOTLEY]
```

Since Fontrix uses screen 1, Joe didn't need to use the A$2000. He made the turtle draw the rock group's insignia beneath the name, saved the completed poster as a Logo picture, and printed many copies for his friends.

Andrew later decided to use Fontrix to put a title on a Koala-and-Logo picture. Since Fontrix readily accepts the Logo picture file format, that was no problem at all.

Students can use the Koala or Power Pad software to fill Logo designs with color. However, they must first decrease the file length and put PICTR. at the front of the filename.

Koala and Power Pad pictures are saved as 33 sector binary files. That is, the picture is stored on 33 small sections of the disk. Logo picture files, on the other hand, need 34 sectors. Thus, students must save their Logo pictures with the following command. (FLOWER is an example of a filename.)

```
DOS [BSAVE PICTR.FLOWER, A$2000, L$1FF8]
```

Then, the graphics software can be used to call the picture for further work. To load it back into Logo once more, follow Andrew's earlier steps.

The use of the DOS command has enriched the Logo activities of my students and colleagues. Why not give it a try yourself?

Terrapin 2.0 Note: The following DOS commands are not supported by Terrapin's 2.0 Logo version: MON, NOMON, VERIFY, INIT, and D2. I have never used the first 3 commands, but I regret losing the ability to initialize disks. The Logo primitive SETDISK allows you to specify which disk drive you want.

For Koala Pad information, write Koala Technologies, 3100 Patrick Henry Drive, Santa Clara, CA 95050.

For information on the Power Pad, write Chalk Board, Inc., 3772 Pleasantdale Road, Atlanta, GA 30340.

Fontrix information can be requested from Data Transforms, 666 Washington Street, Suite 106, Denver, CO 80203.

Nancy Sopp is a special education teacher in the Fairbanks (AK) North Star Borough School District, and has published articles in national educational computing periodicals.

From the Editor

GET SET

There is something special about a beginning, even if it is the beginning of a familiar routine. The competitors at the recent Los Angeles Olympics took their marks and performed magnificent movements which they had practiced for years. They knew the duration of their tasks and the standards for excellence. They knew they must give their best efforts.

At the beginning of the school year, we also take our marks. For the "seasoned veterans" among us, the routine is definitely familiar. We have practiced and trained for this moment. The standards are high - will our best be good enough?

As in past years, the NLX is dedicated to supporting and encouraging your Logo teaching efforts. Our motto, FORWARD 100,

Editor continued

helps us to keep our efforts focused in a progressive and productive direction, and to give you our best - 100%.

As we each approach the school year starting line, let us do so with a quiet confidence that this will be our best year yet with Logo.

Take your marks. FD 100!

Tom

Special Note:

Within the last year, I have become affiliated with several business concerns. I have formed a writing group to produce a series of Logo books for Holt, Rinehart, and Winston, and am currently a consulting editor for them. I have joined the advisory boards of Scholastic, Inc., and Chalk Board, Inc.

In the interest of objectivity, I will not write book or product reviews for the NLX, and shall remove myself from the drafting of statements in the NLX about these and their competitor companies and products. In spite of efforts to the contrary, I may have a subconscious bias toward them, however. If this bias becomes evident in the NLX, please forgive me, and please inform me. Thanks.

Low Logo Prices

Almost Free Software, 78 Main Street, Littleton, NH 03561, sells Commodore Logo for $46.30. (603) 444-2668.

Richard Sundstrom of Strom Systems Inc. announces Terrapin Logo 2.0 version available for $69.95 plus $2.00 shipping. (Michigan residents add 4% tax.) School purchase orders are welcome. Strom Systems Inc., 42303 Ann Arbor Road, Plymouth, MI 48170. (313) 455-8022.

The new Terrapin Logo 2.0 version for the Apple now has a suggested list price of $99.95, and includes a language disk, a utilities disk, and a 520 page manual. The 2.0 version runs on the recently introduced Apple IIc portable computer. In addition, Terrapin offers savings of 47% to 54% when Terrapin Logo is purchased in lots of 10 or 20 copies. Terrapin's new address is 222 Third Street, Cambridge, MA 02142. (617) 492-8816.

Tipps for Teachers

by Steve Tipps

A New Year With Logo

A new year of teaching brings mixed emotions. Will the new group of students be as good (or as unruly) as last year's? Who will ever replace Robert, or Stacy, or Suzanne? And please let me remember how I started last year. Things seemed to start so well! A new year with Logo has a similar mix of anticipation and fear.

*Will the children like Logo this year? Some of the children seemed to get bored by the end of the year. What can I do to keep up their interest and challenge them? I can't think up new ideas every day.

*What if somebody asks me again if they should use " or : before SIZE ? I never have really understood which is which or why; I can finally get it to work if I play with it.

*How can I get children more time on the computer? Those that had a computer at home seemed to make more progress.

*What should I do if a parent says to me, "My child drew circles with Logo last year. I hope you will teach them something useful this year." How can I tell them Logo is not just for drawing circles?

*What if the other teachers think I should teach BASIC this year as an equal opportunity language? What does make Logo so different, special, and important?

These questions loom as real as the first day of school. I can't answer them for you because you must write your own answer procedures. I'll share some thoughts from a summer full of Logo learning. Glen, Tom, and I shared Logo with teachers and parents from all over the United States this summer. We met new and old friends at Logo 84 and NECC. One message repeated throughout the summer was that Logo is changing.

```
REPEAT 1000 [PRINT [LOGO IS CHANGING]]
```

Logo became available commercially only 2 to 3 years ago. Many schools adopted Logo as part of a computer awareness program. Logo seemed like a good "first" language for elementary students. Logo was also fun and helpful in developing computer skills such as saving to disk, loading files, keyboarding and editing. Each new year, new schools are introducing Logo for these reasons.

Tipps continued

Logo is excellent for this limited purpose, but Logo can do much more and you can do much more with it. Teachers and children with one, two or three years of experience are using Logo as a tool to explore the rules, discover meanings, and express ideas in mathematics, language, social studies, art, music, etc. Individual teachers and groups of teachers are looking into content and finding ways for Logo to be useful. A few of the common ideas are:

1. Plotting tools in algebra
2. Graphing tools for social studies and science
3. Sentence and poetry generators in language
4. Perspective tools in art
5. Transposition of keys in music

The content areas are not new; they are standard fare in every school district. Even the activities are not new; you can find programs in BASIC to do the same things. Giving children a tool which they control is new. The message of Logo is that it gives us power with ideas.

Powerful ideas are not restricted to Logo. We must realize that Papert's powerful ideas are powerful because they have power in the world. Logo is a way of playing with variability, patterns, naming, associations, properties, communication and record keeping. Other ways of investigating ideas exist in the science lab or in writing. We have only begun to use Logo.

WHY NOT YET?

Good learning is developmental learning; it takes time. You do not learn how to ride the bicycle, or cook well, or solve differential equations, or be considerate in a single day. Only through experience and guidance do these skills and attitudes develop. If they could be eaten like apples or applied like a poultice, more people would be considerate, gourmet, bicycling mathematicians. Only through many mistakes and gradual improvement, are we able to ride, cook, solve, and be considerate.

One reason Logo is not yet being used completely is that we are still developing skills and attitudes with it. Almost everyone reports that they started with Logo before knowing very much. Any first year with Logo means barely keeping even with the students. It is an exhilarating and frustrating time. To those of you who are beginning, take heart. Keep working and learning, but do not expect to master it in a three hour workshop, or even a three credit course. Logo is developmental and demands time.

Another reason Logo has been limited is that a Logo culture or environment must develop. Dan [illegible]att referred to the "boomtown" era which Logo is in now. The pioneers are there, but Logo needs many settlers to nurture it. Settlers plant Logo in many different places to see where it grows and under what conditions.

What makes Logo thrive in classrooms? Individual teachers working and playing with children find out how it works. What makes Logo work throughout a school? We don't know exactly, but fifteen years of research on innovative programs in math and science indicate at least two things. Constant administrative support of teachers and continual inservice are essential.

Too many schools take the adopt-and-drop approach. School divisions which adopt Logo and drop it on teachers ensure failure. A better plan is to plant Logo with teachers who are willing to spend time to work out the bugs. Logo takes time to grow. Flowering in one classroom will be more effective at creating a culture than dropping Logo like a bomb into three hundred.

QUESTION AND ANSWER

Creating a culture is the present goal of people involved with Logo. The new year's questions deal indirectly with the responses you need to make for yourself in that Logo culture. These questions are asked of me so often that I will try answers for myself. Modify them for yourself.

**Will the children like it this year? What can I do to keep up their interest and challenge them?

If children are still turtling (drawing squares, circles, and doing screen wrap) after a year, they will be bored. Children need to see new things to do with Logo. They also need to think about what they are doing with Logo when they draw polygons.

Here's a challenge for fourth and fifth grade students. Compare the sum of the angles made by the turtle and the sum of the angles inside the figure.

	TOTAL TURTLE TURNS	TOTAL INSIDE ANGLES
square		
pentagon		
hexagon		
circle		

Tipps continued

After working with the turtle, the difference between inside and outside angles seems to blur. There is something to be learned from a well placed question.

You do not have to think up all new ideas. Ideas for Logo are becoming more widely available in magazines and books. The materials at Logo 84 were certainly indicative of an explosion of activity materials. Some are excellent; some reduce Logo to a recipe. But good teachers can take dreadful reading and math materials and lead children into the revels of reading and the magic of mathematical thought.

The teacher who structures the class for discovery has an important ingredient. Look for ideas everywhere. Have an open bookshelf of magazines and books. Put the NLX out for students, as well as Abelson's excellent language manuals. Encourage children to create their own uses. Let children decide what they want to do with Logo next. And let children post challenges for each other. Your task is to set up the environment as a place which invites new ideas.

**What if somebody asks me again if they should use " or : before SIZE ? I never have really understood which is which or why; I can finally get it to work after awhile.

The simple answer deals with the language question. Look at these two examples. "JERRY is a label for a variable. The list in brackets is the value of the variable.

```
MAKE "JERRY [A VERY GOOD SKATER]
MAKE "NANCY [A CHAMPION SWIMMER]
```

You can print "JERRY or the THING about "JERRY.

```
PRINT "JERRY
JERRY
PRINT THING "JERRY
A VERY GOOD SKATER
```

Compare the results to using PRINT with :JERRY.

```
PRINT :JERRY
A VERY GOOD SKATER
```

They are the same. Try it again by PRINTing the THING about "NANCY. Then PRINT :NANCY.

```
PRINT "NANCY
NANCY
PRINT THING "NANCY
A CHAMPION SWIMMER
PRINT :NANCY
A CHAMPION SWIMMER
```

The dots (:) are only a shortened form of THING. Early in Logo, the colon is used for input variables; usually, no explanation is given for why the marks are used. Understanding about THING is the key to understanding which is which and why. That's the easy part.

The harder part of the question is your feeling of confidence. Adults always want to know how to do it and do it right. You shouldn't expect to remember why and how things work in Logo any more than you cook without references. Use resources to supplement your understanding and ask questions. Don't be afraid of making mistakes in front of the students. You can have a bug and say you don't know. You can also demonstrate three or four tries before you get it right.

**How can I get children more time on the computer? Those that had a computer at home seemed to know so much more and make more progress.

Equity in computing is a big issue and likely to get bigger. Many more schools have computers than three years ago, but computing resources certainly do not match Papert's goal for "a computer for every child." This challenge was one of the most controversial ones from Logo 84.

(Ed. Note: The International Council for Computers in Education (ICCE) is becoming actively involved in the equity issue. For more information, write Sharon Franklin, ICCE, University of Oregon, 1787 Agate Street, Eugene, OR 97403.)

Glen Bull suggests that, given a choice of learning with or without paper, he would definitely take paper. But if you only had 2 sheets of paper for thirty students over a full year, you might not be much better off than with none. Thirty minutes once or twice a week is not enough to really dig around with Logo.

Logo is not a recipe for solving all the educational ills of the world. The research results on Logo are inconclusive at this point. Implementation of Logo is extremely varied and likely to stay that way. It is difficult to measure many of the outcomes which Papert hopes to see from Logo use. For methodological as well as philosophical reasons, outcomes for Logo are unlikely in the short run of two years.

Real results are more likely with sustained time on the computer working on content intensive projects. The economics of computing point toward children having a $300 computer system in their lunch pail to take home. My fear is that it will have programmed instruction rather than exploratory

Tipps continued

environments. We are in a transition time when computing is still a precious resource. We should work within our school and communities to make computing more accessible to all students.

** What should I do if a parent says to me,"My child drew circles with Logo last year. I hope you will teach them something useful this year." Can I tell them why Logo is not just for drawing circles?

The parent is justified in asking the question. If students are doing the same projects at fifth grade as at first, Logo is being wasted. The new class may already know as much Logo as you covered last year. But they may not know what they can learn with Logo yet.

Logo does not fit neatly into scope and sequence charts. It operates on the need-to-know principle. Need comes from engagement in meaningful problems. The most personal learning happens when information lets me do something I want to do. Challenge students to write programs which show native American designs, which perform several operations on a mystery number, which graph the results of the presidential election. You will need to find new places to dig with Logo. Look into your curriculum and find projects to do with Logo.

**What if the other teachers think I should teach BASIC this year as an equal opportunity language? What does make Logo so different, special, and important?

Logo is different because it is a language designed for the expression of ideas. Hal Abelson's excellent presentation at Logo 84 on advanced programming pointed out that the most important thing which distinguishes Logo from BASIC and Pascal is not structure or capabilities. The crucial difference is that Logo is an expressive medium, a language to express our ideas.

We make the computer do what we want. Logo is not limited by someone else's restrictions on our thoughts. I have had to learn this lesson many times. If there isn't a certain command which I want to use, I can make it. If a certain procedure doesn't work the way I want, I can fix it. If an idea in social studies isn't clear, I can explore it with Logo.

Logo puts the power of invention in our hands. That's what makes Logo different, special, and important. That's why Logo is a good place to start the new year.

(Ed. Note: Although Steve Tipps has moved to Texas, his column will continue to appear regularly in the NLX. We wish him the best in his new position! Write to him in care of the Education Department, Midwestern State University, Wichita Falls, TX 76308.)

Steve Tipps is the West Professor of Education at Midwestern State University in Wichita Falls, TX. He presents Logo workshops for school systems throughout the United States.

前 100

by
Hillel
Weintraub

Greetings from Nippon, the land of sushi, Koji Gushiken, and a growing Logo community. Each month, I'll share some aspects of that community in this column: the hardware, the software, the people, and the ideas.

By the way, the Japanese character above is pronounced something like "Mah-eh." I'm sure you can figure out what it means. You have just learned to read your first Japanese kanji without anyone "teaching" you!

A LITTLE BIT OF NEWS

This month, I want to tell you about a new notebook sized computer from Japan, the Epson HC-80. It has a Z-80 microprocessor, 80 column by 8 line liquid crystal display (LCD), a microcassette storage device, and a price tag of around $1000. A Wordstar cartridge is available to make it into a rather nice little system.

But here's the exciting news. Epson is working on a Logo version for the HC-80! It "should" be available before the end of 1984. This would make it one of the most portable Logo versions ever. Think of the application possibilities!

With computers becoming smaller and more powerful, the HC-80 certainly has the shape of the computer of the future. Although not yet available in the US as of this writing, the HC-80 and its Logo could become the "take home" notebook of tomorrow's students.

If you have questions about Logo in Japan, please write to me at Doshisha International High School, Tatara, Tanabe-cho, Tsuzuki-gun, Kyoto-fu 610-03 Japan.

Hillel Weintraub is a teacher at Doshisha International High School in Kyoto-fu, Japan, and is the president of the Society for Microcomputing in Life and Education (SMILE).

Listful Thinking

by
Glen Bull and Paula Cochran

Poetry Sparks

Procedures which operate on lists are pieces of Logo machinery, just as spark plugs and carburetors are pieces of automotive machinery. In programming class, it is important to learn how to build these procedures. In English class, it is more important to use them, for example, to create poetry.

THE IDEA OF EXTENSIBILITY

An English teacher cannot spend great amounts of time teaching programming skills before using the computer to create poetry. In Logo, it isn't necessary.

Logo allows a teacher to add new commands to the language. This quality is called extensibility. Extensibility opens the door for creation of a Logo environment in English, math, and other content areas. An English teacher using Logo can supply list procedures which students use as if they were built-in commands. The procedure PICK can be used this way.

PICKING THE RIGHT WORD

PICK randomly selects a word from a list. Type this procedure:

```
TO PICK :LIST
OUTPUT ITEM 1 + (RANDOM COUNT :LIST) :LIST
END
```

(**Note: ITEM and COUNT are not built-in commands in Version 1 of Terrapin and Krell Logo for the Apple. If necessary, see the end of the column for these procedures.)

Now try PICK with this list of words:

```
PRINT PICK [MILK EGGS APPLES BREAD BUTTER]
BREAD
```

What did PICK choose when you tried it? Use PICK with your own list of words:

```
PRINT PICK [ ____________________
```

Here is how to use PICK inside another procedure. This procedure picks a poet.

```
TO CHOOSE.POET
OUTPUT PICK [POE LONGFELLOW SHELLEY FROST]
END
```

Now use the CHOOSE.POET program by typing:

```
PRINT CHOOSE.POET
FROST
```

Two guidelines will help you use PICK inside another procedure:

(1) When PICK is used inside a procedure, place the command OUTPUT in front of PICK.

(2) When PICK chooses a word, you must tell Logo what to do with the word. In the example with CHOOSE.POET, Logo is told to PRINT the word chosen (FROST).

CREATING A SENTENCE

Use CHOOSE.POET to make a sentence.

```
PRINT SENTENCE [I LIKE TO READ] CHOOSE.POET
I LIKE TO READ SHELLEY
```

What did you get when you tried it?

```
I LIKE TO READ ________________
```

The command SENTENCE is built into Logo. It puts two lists of words together into a sentence. The Logo abbreviation for SENTENCE is SE.

```
PRINT SE [HAIL TO THEE,] [BLYTHE SPIRIT!]
HAIL TO THEE, BLYTHE SPIRIT!
```

BULIDING A POEM

Use PICK to choose rhyming words in a poem:

```
TO POEM
PRINT SE [THE FASTEST FUNNIEST] RHYME.A
PRINT SE [WAS ALMOST NEVER] RHYME.B
PRINT SE [HIS FRIEND THE BIG BAD] RHYME.A
PRINT SE [WAS ALMOST ALWAYS] RHYME.B
END
```

RHYME.A and RHYME.B are Logo procedures which pick rhyming words for the end of each line.

```
TO RHYME.A
OUTPUT PICK [CAT GNAT BAT RAT BRAT]
END
```

Listful Thinking continued

```
TO RHYME.B
OUTPUT PICK [GREEN MEAN KEEN CLEAN LEAN]
END
```

Here is an example:

```
POEM
THE FASTEST FUNNIEST RAT
WAS ALMOST NEVER LEAN
HIS FRIEND THE BIG BAD CAT
WAS ALMOST ALWAYS MEAN
```

WHAT IS LEARNED?

Use Logo poetry to explore literature, grammar, and vocabulary. A literature class, for example, might want to explore rhyme schemes. Our sample poem has a rhyme scheme of A B A B. Students could use PICK and SE to create a limerick, for example, a rhyme scheme of A A B B A. Other poetry forms, such as sonnets, have even more complex patterns which can be explored with Logo.

Our poem also has a grammatical pattern. The pattern of word categories is:

The fastest funniest	NOUN
Was almost never ...	ADJECTIVE
His friend the big bad	NOUN
Was almost always ...	ADJECTIVE

When nouns and adjectives are mixed in a rhyme list, the presence of word categories is discovered:

```
THE FASTEST FUNNIEST GREEN
WAS ALMOST NEVER CAT
```

Instead of memorizing the properties of nouns and adjectives, students can discover these properties by using them in a playful way.

HOW MUCH PROGRAMMING?

The point is, you don't have to teach Logo to use Logo -- in English class, or any other class. It isn't necessary for each student to develop their own PICK procedure, because the teacher can provide it. This allows the class to focus on poetry instead programming. Here are some of the things a teacher might supply to create a Logo environment for poetry:

(1) Logo tools such as PICK

(2) Some examples or models of how PICK can be used

(3) A content-related project, such -- writing a limerick, sonnet, or ballad.

Hal Abelson suggests that a good Logo environment provides a way to express ideas, in the same way that paper provides a way to express ideas. What must an English teacher give up when her students use Logo? Nothing more than she gives up when her students use paper and pencil.

-+-+-

Here are the procedures required to use PICK with Version 1 of Terrapin and Krell Logo.

```
TO ITEM :NUMBER :LIST
OUTPUT ITEM.LOOP :NUMBER :LIST 1
END

TO ITEM.LOOP :NUMBER :LIST :COUNT
IF :LIST = [] OP (SE [THERE ARE ONLY]
 :COUNT [ITEMS IN THIS LIST.])
IF :NUMBER = :COUNT OP FIRST :LIST
OP ITEM.LOOP :NUMBER BF :LIST :COUNT + 1
END

TO COUNT :LIST
OP COUNT.LOOP :LIST 0
END

TO COUNT.LOOP :LIST :NUMBER
IF :LIST = [] OP :NUMBER
OP COUNT.LOOP BF :LIST :NUMBER + 1
END
```

Glen Bull is a professor in University of Virgina's School of Education, and teaches Logo courses at both the graduate and undergraduate level. Paula Cochran is a reformed English major who studied linguistics at the University of Cambridge. She is now a speech-language pathologist working with language-disabled children.

TV Turtle

Logo is on television! The NABU Network Corporation of Ottawa, Ontario, has been broadcasting a weekly show called "The Turtle Club" to its cable subscribers. The program features instruction, lively discussions, and demonstrations of Logo procedures. The network also provides its own Logo version (similar to Sprite Logo for the Apple) which is downloaded into NABU computers via cable. NABU now broadcasts into Alexandria, VA, and is scheduled for 20 additional US cities soon. For more information, write Ian Rae, NABU Logo, 1051 Baxter Road, Ottawa, Ontario, K2C 3P2, Canada. (613) 596-6700.

Apple Logo II:

It's Bigger, Better

by
Jason Gervich

Logo lovers, rejoice! The Logo you've been waiting for has finally arrived. At the Apple II Forever show this past April, Apple announced its new Apple Logo II.

Logo II is a powerful and complete programming language designed by Apple and LCSI. It runs on the 128K configuration of the Apple IIe and the Apple IIc computers. When booted, Logo II boasts a whopping 7186 nodes of workspace. No more OUT OF SPACE messages (at least for a while)! The version also features upper and lower case letters, 40 or 80 column text mode, and an enhanced editor.

In addition, Logo II contains 30 file handling commands. These turn Logo II into an all purpose computer language. It is easy to create dribble files, data files, picture files, and random access files. Included also is a "print using" primitive called FORM which takes three inputs (number, field, and precision) and allows professional displays of financial calculations.

Other enhancements include:
*Debugging: TRACE and STEP
*Music: TOOT
*Graphics: SAVEPIC, PRINTPIC, and FILL
*Machine Language: .CALL
*Line Detection: DOTP

Practically everything in the earlier Apple Logo version is kept or enhanced in Logo II. One exception is the PACKAGE command. It has been replaced with more flexible BURY commands. You can now bury and unbury procedures, names, or the entire workspace, and save selected groups of procedures to a disk file.

WHAT YOU GET

Logo II comes in an attractive bookcase styled box, with a double sided training disk, the Logo II language disk, and an introductory tutorial and reference manual.

The training disk is interactive, and seems to be an excellent tool for introducing Logo to beginners. It contains lessons on saving and loading procedures, using the editor, and working with variables and lists, as well as turtle graphics.

The improved reference manual contains 325 pages, including sections on file handling and the new Apple disk operating system, ProDOS. The use of ProDOS allows Logo II to be used on a hard disk in a network.

ProDOS does not recognize disk files from the earlier Apple Logo version until converted with the ProDOS conversion utility, a straightforward task.

The language disk contains a plethora of tool kit procedures, including graphics, arithmetic, debugging, and list processing tools. In addition, a set of procedures and documentation is provided for support of the Apple Mouse. With a little work, the mouse can be configured for use as a pointing device for text or menu selection, or as a graphics device for manipulating the turtle.

A HELP primitive displays the inputs to any procedure, and provides several HELP screens for on line guidance. Customized HELP screens can be added for specific teaching situations.

Logo II loads picture files quickly (about 3 seconds), thanks to ProDOS. Further, pictures created with the Mouse Paint program can be brought into the Logo II environment for further work or use. (Logo II picture files are also recognized by Mouse Paint, by the way.) Just think of the possibilities for title screens, drawings for adventure games, and the like!

(Ed. Note: See Nancy Parks Sopp's DOS article in this issue for more ideas on bringing pictures drawn with other programs into a Logo environment.)

Logo II does not facilitate operations with parallel devices, such as printing with a parallel printer, because it does not recognize the Apple Parallel Card in the Apple IIe. It may work with other models, however. Test before you buy. Operations with the Apple Super Serial Card are supported.

IN CONCLUSION

Apple Logo II is a powerhouse of a language which may redefine the Logo experience. Its implementation on a 128K machine releases much of the power that Logo lovers have been waiting for, and helps bring the nongraphics side of Logo closer to the light.

Jason Gervich is a Logo enthusiast from Palo Alto, CA, and has written computer articles for several national publications.

Micro City, PO Box 571, Downers Grove, IL 60515, offers the Sweet P plotter for $479. Call (312) 355-9726.

Results of NLX Logo Survey Announced

Here are the results of the NLX Logo Survey conducted in April and May.

++Logo use by grades. 1677 grade responses.

Grade	Percent
Kindergarten	8.17
First	9.12
Second	10.97
Third	13.18
Fourth	13.95
Fifth	13.89
Sixth	11.63
Seventh	5.78
Eighth	5.49
Ninth	2.74
Tenth through Twelfth	2.21
Post-secondary	2.87

Here is another grouping.

Range	Percent
K-2	28.26
3-6	52.65
7-9	14.01
10-12	2.21
Post-secondary	2.87

++Logo version use. 580 version responses.

Version	Percent
Apple Logo	40.00
Terrapin Logo	31.72
Krell Logo	8.97
Texas Instruments Logo	4.48
Commodore Logo	4.48
Atari Logo	4.14
Dr. Logo	2.24
TRS-80 Color Logo	2.07
IBM Logo	1.38
Other	0.52

++Logo book use.

Dan Watt's Learning with Logo completely dominated the survey. Others mentioned included Harold Abelson's Apple Logo and Logo for the Apple II, Gary Bitter and Nancy Watson's Apple Logo Primer, and MECC's Introduction to Logo for Teachers.

Special thanks to all who sent in the blue cards so promptly. Thanks also to Kevin McCurdy for tabulating the responses.

Logo Disserts:

Dissertations Dealing with Logo

by Barbara Elias

Can the Logo environment influence young children's cognitive style? This question was examined in L. M. Young's 1983 study which investigated the effects of the Logo learning environment on the Reflexive / Impulsive cognitive styles of second grade children.

Current research supports the notion that certain cognitive processes appear to support or hinder good problem solving. The Reflexive / Impulsive dimension of cognitive style concerns behaviors in children related to the amount of time a child spends deliberating on solutions in problem solving tasks. Generally, children who are reflexive perform better on visual discrimination tasks, serial recall, inductive reasoning, and reading.

After being classified based on Matching Familiar Figures Test: Child scores, 12 impulsive students were placed in a Logo learning environment for 15 one hour sessions during a 3 week period. Posttest results with the same instrument showed no significant differences in the experimental group. However, more of the experimental group than the control group shifted toward reflexive behavior.

From a practical standpoint, Young noted that all of the children experienced success in programming the computer and in designing programs. Given more time, the type of learning environment provided by Logo may aid children in becoming more self-conscious reflexive thinkers.

>>>Young, L. M. (1983). An Analysis of the Logo Computer Programming Environment upon the Reflexive and Impulsive Cognitive Styles of Second Grade Students (Doctoral dissertation, University of Pittsburgh, 1982). Dissertation Abstracts International, 44, 64-A. University Microfilm No. 83-12545.

(Ed. note: Each month, Barbara Elias highlights a dissertation or thesis dealing with Logo. She is accepting copies of recent research results for the NLX Dissertation and Thesis Repository. Reports may be mailed to her in care of NLX, PO Box 5341, Charlottesville; VA 22905.)

Barbara Elias is an assistant professor in the Education Department of Virginia State University in Petersburg, VA, and a doctoral candidate at the University of Virginia.

Q and A

by
Jim McCauley

Q. I am organizing a Logo course for teachers who already have had some Logo experience. Could you suggest an activity for the first class meeting to get us off to a good start?

Q. Until recently, I thought that the Logo primitives were commands. A friend told me that some of them are called operations. Can you help me to understand the difference?

<[-?-]>

A. My comments this month combine the responses to both of the above questions. I'll tell you about an exercise I have found useful as a kickoff for a course for experienced Logo teachers. Not everyone likes it, but it succeeds in doing two things: insuring that those taking the course become familiar with the range of primitives in Logo, and convincing everyone that I am serious.

DOERS AND PROVIDERS

Before I assign the exercise, I familiarize the class with the difference between commands and operations. (Both commands and operations are primitives, by the way.) Commands always cause something to happen. For example, if I type FORWARD 100, the turtle moves forward. If I type CLEARTEXT, all text on the screen disappears. You can think of commands as "doers".

Operations are different. An operation is a "provider of information". For example, XCOR provides a number that represents the left-to-right position of the turtle on the screen. FIRST provides the first item in a list, the first letter in a word, or the first digit in a number. The most important difference between an operation and a command is that an operation outputs information to a command or some other operation.

HOW TO TELL THE DIFFERENCE?

There are several ways to tell whether a primitive is a command or an operation. The easiest way is to look it up in the reference manual. If the description mentions that the primitive in question outputs something, then it is an operation. If output is not mentioned, chances are that it is a command.

Another way to find out is to sit down at the computer and see what happens when you type in the name of the primitive. If you get a response such as NOT ENOUGH INPUTS TO (something), in LCSI Logo, or (something) NEEDS MORE INPUTS, in MIT Logo, it means that the primitive cannot act until the primitive name is typed in followed immediately by the right sort of information. This response is typical for a Logo command which needs input. For example, have you ever typed FORWARD without a number?

Operations, on the other hand, give a response such as I DON'T KNOW WHAT TO DO WITH (something), in LCSI Logo. In MIT Logo, the response is RESULT: (something).

WHY BOTHER?

It is important to understand the difference between commands and operations for at least two reasons. One is simple: Every line in a Logo procedure must begin with a command. If the line begins with an operation, the procedure does not function, because there is no command to accept the operation's output or to do something with it.

The other reason is not so simple. Most people are introduced to Logo through turtle graphics, a wonderful way to begin programming computers. However, most of the Logo procedures written by beginners contain only commands. Logo operations are rarely introduced in a first course. This is unfortunate, because many of the most powerful features of list processing depend on a good understanding of operations and how they output information.

HOMEWORK ASSIGNMENT

When I assign the exercise, I divide the class into groups of four. Each group member is assigned the task of compiling a list which contains one of the following. No two group members compile the same list, of course.

1. All the Logo commands that require information. Examples include FORWARD SAVE OUTPUT
2. All the Logo commands that do not require information. Examples include CLEARTEXT PENDOWN STOP
3. All the Logo operations that require information. Examples include FIRST SENTENCE TEXT
4. All the Logo operations that do not require information. Examples include XCOR HEADING READLIST (or REQUEST)

Q and A continued

Once the effort is over (and it is a major effort), the group members share their findings. They discover that they have acquired a good understanding of the difference between commands and operations, as well as an appreciation of the range of things that can be done in Logo.

Let me close with this question for thought. Is the primitive IF a command? An operation? Both??

(Ed. note: If you have questions about Logo programming techniques or the thinking which leads to Logo programming, send them to NLX Q & A, Post Office Box 5341, Charlottesville, VA 22905. Each month, Jim McCauley answers selected questions in this column.)

Jim McCauley is a Coordinator of Computer Education for the Santa Clara (CA) County Office of Education, and has written Logo articles for several national publications.

Teacher to Teacher

Logo Guided Discovery Kits

reviewed by

Ann Cairns Federlein and Jim Fry

Teachers are always looking for good materials to expand the learning activities in their classrooms. The "Logo Guided Discovery Kits," Apple Logo version, recently published by Microcomputer Resources, is a find for the teacher who is new to teaching Logo in the early elementary grades.

WHAT IS IN IT?

The kit comes in two versions. Kit 1+ is for use in lower grades. Kit 3+ is designed for upper elementary grades. Both kits contain discovery cards, reference cards, duplicating masters, turtle trainer buttons, a facilitator manual, and keyboard and command posters. Kit 1+ also contains a well constructed turtle hand puppet. Both kits include ideas for use with elementary aged children to stimulate creative thinking.

The facilitator's manuals are well organized and contain an introductory section, a scope and sequence, discovery card support materials, and a bibliography. The introduction gives a great deal of valuable information on the content and structure of the kit, the role of the facilitator, and the effective use of questions. Ideas on guided discovery and learning modalities are included, as well as tips for setting up and managing a Logo environment.

The scope and sequence includes areas such as problem solving, mathematics and geometry, Logo programming, and computer literacy. Kit 1+ also concentrates on early childhood development and concept development.

The discovery card support materials section contains a picture of each student card, along with student objectives, helpful hints, materials required, pre and post activities and questions, and enrichment activities for students and facilitators who want to go further. All information for each card can be seen without having to turn a page.

HOW TO USE IT?

Given the pressures in today's classrooms, teachers might be tempted to purchase a kit which has all the lessons in sequence and just follow the suggested guide. The kits were designed to be aids for teaching Logo. They are not the only way. They give teachers a starting point to develop their own lessons based on the needs of their students.

How Logo teachers use the kit activities makes the difference between children just memorizing commands and learning them. The abundant number of duplication masters in the kit is a concern; pencil and paper tasks foster drill and practice exercises rather than emphasizing the understanding of Logo principles.

To take advantage of Logo's potential, kit activities should be presented so that children are "guided" to discoveries in Logo, not just "directed" to accomplish the assignments. Many activities encourage peer teaching. It is the teacher who determines whether students understand the process of Logo, or simply follow directions.

The Logo Guided Discovery Kits appeal to young children. The colorful aids and large, easy to follow charts are durably constructed, allowing for many hours of classroom use.

The list price of Kit 1+ is $159.95, and Kit 3+ is $139.50. Microcomputer Resources, 2845 Temple Avenue, Long Beach, CA 90806. (213) 424-0763.

Questions and comments on this review should be sent to Logo Information For Teachers (LIFT), 37279 Six Mile Road, Livonia, MI 48152.

Anne Cairns Federlein is a professor in the Early Childhood Area, School of Human and Educational Services, Oakland University Rochester, MI. Jim Fry is an elementary teacher working with first through fourth grades in the Novi Community School District, Novi, MI.

TurtleTips

by
Donna Lanyi
and Jane Toth

When we first started using Logo in the classroom, we needed and searched for beginner teacher and student activities. Many of the articles and activities we found were geared to more advanced Logo users. This column is dedicated to monthly procedures, teacher hints, and ideas for the novice. Our ideas will follow a theme centered around holidays, seasons, and specific events.

TEACHER HINTS

In getting our students ready for Logo activites, we've discovered some useful tactile / kinesthetic activities to practice beforehand. The most popular game has been "Playing Turtle." One student plays the Logo turtle following exactly the directions given by another student, the programmer. The programmer tries to get the turtle to a designated spot.

Other options for "Playing Turtle" are the teacher playing turtle with students giving directions, a student or students playing turtle with the teacher as programmer, or giving the turtle a tail made from a yardstick with a marker or chalk attached so a visible path can be left on the sidewalk or paper.

Another pre-computer activity is to have students draw on a sheet of grid paper following directions given by the teacher. Primary teachers could have students place a paper turtle over their pencils before beginning the activity. A HOME position and the size of the grid need to be predetermined. An easy checking method is created because a quick glance among the group allows the teacher to see whose pattern is not the same. Later, children can practice and explore with designs of their own.

"Turtle Says," a version of "Simon Says," can also be played. Students follow turtle commands in the same manner as "Simon Says."

PROCEDURES OF THE MONTH

"Lost Turtle" and "Turtle Target Practice" are two procedures that allow students to gain experience in moving the turtle on the screen. Both games randomly place targets on the screen. Students need to move the turtle in the least number of steps to "win". If you are interested in a listing of these procedures, please send a stamped self addressed envelope to the address listed below.

We invite teachers and elementary students to send their procedures or ideas to be shared. Send to Donna Lanyi & Jane Toth, Beall Avenue School, 716 Beall Avenue, Wooster, OH 44691.

Donna Lanyi and Jane Toth use Logo in their elementary classrooms in the Wooster (OH) City Schools.

NLXionary

A Lectionary of Selected Logo Readings

by Griff Wigley

This month, six of the seven selections are from publications devoted to IBM computers. Most of the articles have relevance for any Logo user, however.

PCjr Magazine, June 1984, "Looking at Logo," column by Winn L. Rosch

>The second of a two part interview with Seymour Papert. The May column concentrated on Logo's implementation on the IBM PCjr. This part focuses on Logo's role in elementary education. Papert talks empathetically about teachers, explains his personality trait theory on children attracted to turtles vs. sprites, and takes his customary pokes at the structure of public education.

PCjr Magazine, August 1984, "Looking at Logo," column by Winn L. Rosch

>A nifty mortgage amortization program in Logo is explained, demonstrating Logo's serious nature. This would be a perfect Jim McCauley type project, if presented in a way to encourage the reader to tackle the solution to the problem, instead of the "let me explain my solution" approach.

PCjr Magazine, September 1984, "Looking at Logo," column by Winn L. Rosch

>A brief explanation of six publications devoted to Logo and the people behind them: Dr. Logo Newsletter, Logo and Educational Computing Journal, Polyspiral, Turtle News, TurtleTalk, and the NLX. No mention of Logophile or FOLLKlore.

NLXionary continued

jr Magazine, July 1984, "Learning Logo," column by James D. Milojkovic

>A very concise, friendly, and innovative introduction to the programming, learning, and philosophical perspectives of Logo. This article is a standout, not only for newcomers, but for all who have been involved with Logo for a while and need a refresher on what we're really supposed to be up to. The author is billed as a "cognitive psychologist engaged in research on the effect of microcomputers on learning and motivation."

jr Magazine, August 1984, "Learning Logo," column by James D. Milojkovic

>A full explanation of the random "Mondrian" program originally put in the public domain by LCSI. Also included is a diagram of the "procedure calling trees," a feature rarely demonstrated in Logo articles. Many advanced Logo features are explained, appropriate for those in the medium ability range who want to raise their level of sophistication with the language.

Byte Magazine, August 1984, "Four Logos for the IBM PC," by Mark Bridger

>Byte's annual language issue is focused on Modula-2, but contains this technically oriented comparison review of Dr. Logo (Digital Research), IBM Logo (LCSI), PC Logo (Harvard Associates), and Waterloo Logo (Waterloo Microsystems). The author gives a slight edge to IBM Logo over PC Logo for first place, with Dr. Logo a close third, and Waterloo Logo fourth. However, the explanations for his ranking led me to a somewhat different conclusion.

PC World, July 1984, "IBM Rides the Turtle," by Greg Stone

>A straightforward review of the features and shortcomings of LCSI's IBM Logo. Stone's conclusion: "IBM Logo is clearly better than Waterloo Logo, but when you start comparing it with PC Logo and Dr. Logo, you have to decide what features are most important to you." Before you buy Logo for your IBM PC or PCjr, read this article, the Byte article, and Reinhold Wappler's "Logo for the IBM Report."

(Ed. Note: For a free copy of the Wappler Report, send a long, self addressed envelope with 54c in US postage or postal coupons to IBM Report, c/o NLX, PO Box 5341, Charlottesville, VA 22905.)

Griff Wigley is a co-owner of Family Computing, Inc., a facilitator in the Faribault (MN) Public Schools, and a school board member of Prairie Creek Community School.

Logo 84
Held at MIT

Here are some personal comments and reactions to the Logo 84 conference.

Andrew David:

What happens when an artifact hatched in a laboratory environment leaves home in a world of many philosophies, objectives, and imperfect scenarios? With so many voices beginning to be heard in an ever widening range of experience, the first national Logo conference at MIT this past June was timely.

Who was there? Teachers, curriculum designers, computer scientists, network organizers, researchers, developers, administrators, and video producers. 16 countries and 6 continents were represented. More than half of the participants were women, something that would have been unheard of in a computer conference 10 years ago.

What were they talking about? Everything. Open classrooms in the 60's, kinesthetic game rooms in the 90's; what is possible versus what is important; proper methods for evaluating Logo learning; supporting Logo in a classroom with a computer shortage; Logo and the hearing impaired; Logo in business; even Logo in religion.

One of Papert's main points was that we must search for ways to present knowledge in reconstructible fragments, an embellishment of his "mind sized bites" idea. Like the children he would have us reach, we at the conference played with fragments and reconstructed them through dialog.

Where do we go from here? Logo is still going strong. Newer more powerful versions will come, but we don't know when. Support material is trickling in. We still don't know what a microworld is. Clearly, there are far more new Logo users than people who can support themselves for now, at least. Logo's growth will depend on the initiative of creative activists. But an international family has been established. That will count for a great deal.

Andrew David is an educational communications consultant working in the Chicago area. His Logo activities include advanced programming, workshops, and teacher training.

Logo 84 continued

Ihor Charischak:

Logo 84 inspired me. There was something special about this conference - something that I had not experienced before. As I looked back over those three days to find what it was that moved me, I realized that it evolved from my participation with the people who were there.

These people were, for the most part, educators. A few were vendors. Some were students, and the rest were just plain Logo enthusiasts. But I have been to other meetings where similar people had gathered, and didn't walk away so moved.

The difference here was these people had something in common - a vision of what formal education could look like. They were not interested necessarily in bigger, better, or different versions of the same thing, but in establishing a learning environment that was fundamentally different. They called it the Logo environment.

The question that was raised frequently during the conference was, "What exactly is the Logo environment?" Many people described its characteristics, but no one seemed to capture its essence completely.

The truth may be that this environment is not possible to define because it is a spirit rather than a thing - something that can only be satisfying if experienced, not languaged. Without the experience, the words themselves do not make a difference.

I discovered in Cambridge that the essence of the Logo environment is love. Its manifestation did not take on any specific form. I experienced it when I shared with others my vision of the impact that Logo can make on children's lives. Other people may have experienced it by sharing with a large audience in an auditorium, or with a friend over coffee.

What we did at Logo 84 is support each other to go out in the world and model to our colleagues and friends our unique vision of how we can create communities of learners where the motivation is love. To me, that is the key to making education work for all on this planet.

Ihor Charischak directs the Board of Cooperative Educational Services (BOCES) Southern Westchester Instructional Services Department in Ardsley, NY.

Tom Lough:

Logo 84 was a special time. Over 500 enthusiastic participants enjoyed 3 days of panels, group discussions, poster sessions, and other sharing opportunities. There were a number of phrases which seemed to characterize many of the discussions.

"Logo culture" was heard many times, from Dan Watt and others. The role of such a supporting culture in learning with Logo is crucial. As teachers, we must recognize this role, and do our best to establish and maintain this nurturing environment.

Seymour Papert used the phrase "fractured knowledge" in describing ways people learn. We assimilate bits and pieces of information from here and there, eventually putting it all together in a way that makes sense to us. Learning with Logo is often like that.

"A medium for expressing ideas" was used by Harold Abelson to describe a family of computer languages which include Logo, LISP, Smalltalk, and Prolog. This characteristic is absent from most other languages. The ability of Logo to handle and display generalizations of patterns and ideas hints at educational power which is, for the most part, untapped as yet.

"What is the effect of Logo on ——?" is a typical educational research question, and stimulated many animated discussions regarding Logo research. Brian Harvey provided an interesting perspective by reminding us that the pretest / posttest statistical methods were based originally on agricultural experiments. He pointed out that the researchers could not ask an ear of corn how it felt; they also did not care about any individual ear of corn. This contrasts remarkably with the educational environment.

"Extra copies of the pre-proceedings" was a concern of many. Although each participant received a copy, it was evident that this significant document deserved much wider distribution. Its 200 pages contain 17 panelist papers, abstracts of 50 poster sessions, and a bibliography listing over 70 Logo books and hundreds of Logo articles. There are a limited number of copies of the Logo 84 pre-proceedings still available. To obtain a copy, send a check (made out to MIT Logo Group) for $15 US to MIT Logo Group/Publications, NE43-438, 545 Technology Square, Cambridge, MA 02139.

"When is Logo 85?" everyone wanted to know. It is already in the planning stages. Send your ideas and suggestions to the MIT Logo Group at the above address.

NLXual Challenges

by
Robs Muir

It's About Time

A wise man once said that the educational value of Logo lies not in the way it will affect children, but, rather, on the effect it will have on teachers.

The NLXual Challenges are intended to support that community of teachers interested in experiencing the same frustrations of learning and joys of discovery that they encourage in their own classrooms -- rooms rich in problem solving opportunities. Learning is important at all ages, and turnabout is fair play!

ON TASK

The task of this column is simple and serious; a problem is posed which is yearning for a Logo solution. Each problem will exercise a different facet of your Logo "interpreter". Your mission (should you decide to accept it) is to figure out how to solve said problem using whatever version of Logo you have.

When you have a working solution (or even a valiant attempt), share it with the Logo community. We compile the contributed solutions, select the most interesting and unusual, and distribute them to each contributor.

Since our "community" now encompasses the US, Canada, and several overseas countries, the contributions are quite diverse. Some possible solutions are available to each problem, but you must contribute your attempted solution (however meager) to receive these "answers". Send your solution, along with a self addressed stamped envelope, to NLXual Challenges, Attn: Robs Muir, 1688 Denver Avenue, Claremont, CA 91711.

TAKE THIS

September's problem: Create a text based digital clock that will display hours, minutes, and seconds. A suggested format might look something like one of the following.

```
Hours: 7  Minutes: 23  Seconds: 45

7:23:45

7 Hrs. 23 Min. 45 Sec.
```

Accuracy is encouraged, creativity is applauded, complexity is discouraged, and excessive use of global variables is a sure sign of a BASIC programmer. If you are a more sophisticated user, can you improve your clock's accuracy by avoiding periodic "garbage collection" (.GCOLL or RECYCLEing)?

For those of you unfamiliar with the Logo textscreen primitives, here are some words to explore: CLEARTEXT, CURSOR (SETCURSOR), LOCAL, NODRAW, PRINT, PRINT1 (TYPE), SENTENCE, TEXTSCREEN, and TECHNICAL MANUAL (REFERENCE MANUAL).

The month of September is ticking away...

Robs Muir is a teacher in the Claremont CA Unified School District and Claremont Graduate School. He presents Logo workshops in the southern California area.

Kaypro Logo

In the July / August issue of Profiles, the magazine for Kaypro computer users, an ad by MicroSphere, Inc., appears on page 7, announcing a Logo version for their color graphics board for the Kaypro portable computer!

For more information, write MicroSphere, Inc., PO Box 1221, Bend, OR 97709, or call (503) 388-1194.

The National Logo Exchange, copyright (c) 1984 by Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly, September through May, $25 (US) per year, mailed first class from Charlottesville, VA. $5 (US) additional per year for addresses outside the USA, Canada, and Mexico. The opinions expressed by the authors are not necessarily those of The National Logo Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress Street, Salem, MA 01971. Address other correspondence to: The National Logo Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor.................. ....Tom Lough

THE NATIONAL LOGO EXCHANGE

Volume 3 Number 2 FORWARD 100! October 1984

Summer Fun with Logo

by Mary Bonness

I felt an overwhelming sense of panic when I discovered that my summer computer class had students ranging in age from 6 to 12 years old! My brain circuits seemed to overload, then overflow with doubts, questions, and fears.

How could I possibly challenge the older students while simultaneously leading the younger students at an introductory pace? What type of battle plan would provide for the survival of us all?

Somehow, we made it through, with surprising success. I'd like to share what happened, in hopes that some ideas might be useful to others.

TAKING STOCK

Nine computers were at my disposal with an average class size of twelve. Not bad. The students were from different schools with a variety of experience. I quickly discovered that there was a common ingredient of ENTHUSIASM!

I took advantage of the excellent materials available for helping with Logo teaching. The Turtle's Sourcebook by Kathleen Martin and Donna Bearden was invaluable, as well as Turtle News, the newsletter of the Young Peoples' Logo Association. Harold Abelson's Apple Logo contains numerous graphic examples which proved to be excellent starter challenges for students.

OFF COMPUTER

I was surprised to discover how popular, not to mention how helpful, the off-computer activities were with the students. Using ideas from the NLX and the other references, I tried to start each lesson with something concrete.

The students loved to take turns playing turtle. Blindfolded, they were directed by their classmates through a maze of desks or along some specific path in the room. The only instructions they could follow were turtle commands.

Marking the turtle's trail on the floor with masking tape was a wonderful graphic way for the students to see and walk through the geometric forms.

The younger students designed a turtletown and took turns steering their walnut shell turtles to the candy store, the farm, the swimming pool, and the theater. I mistakenly thought the older students would consider this babyish, but they were eager to drive their turtles around town also.

ON COMPUTER

The students were given a specific assignment at the beginning of class, usually some type of graphic design which would require the use of a particular command or concept. After solving this problem, they were free to work on other projects of their own choosing.

The different age levels seemed to complement each other, and there was a great deal of sharing. At times, the younger students would create a fantastic pattern by "accident," and cause the older students and me to come scrambling. This taught us all something new.

The mixture of off and on computer activities seemed to work well. It was delightful to teach students who groaned with agony when class was finished. (I groaned, too!) Time always seemed to evaporate. Discipline problems were minimal. After all, when the students are practically glued to their monitors, what else is more interesting?

At the end of the summer session, each student worked on a SHOW.OFF program which incorporated all their favorite procedures into one fancy production. I was amazed and delighted to see their creations.

Summer continued

A VITAL ELEMENT

The learning was non-stop and the enthusiasm was contagious. I found that the element of discovery is vital for this spirit. It is important for the teacher to step back and let the students explore, without providing a complete map for them.

This summer experience was very enriching for me. Although I felt like I was diving into icy cold, unknown waters over my head at first, I found the rewards were well worth the risk.

Mary Bonness teches at The Greenhill School in Dallas, TX, and conducts Logo workshops for teachers.

From the Editor

Twenty/Twenty and More

Well, it had to happen sooner or later. Like so many others before me, I wasn't ready when it came in the mail - the invitation to my 20th college reunion. As the song says, "Yesterday came suddenly." Why do I feel so old?

It couldn't have been twenty years ago that I sat down in front of the computer with all the flashing lights and nixie tubes for my first lesson in BASIC. The console and accompanying components filled most of the room. I had to punch in my instructions on a teletype. But, it worked, and it hooked me.

Some years later, I "graduated" to FORTRAN on an IBM 360 mainframe computer. I remember spending hours on keypunch machines, hours waiting for results, and more hours repunching culprit cards.

Today, I find myself using Logo on an Apple or IBM PC which can run circles around that early beast. Amazing. It is difficult to appreciate fully the technological progress of the last two decades.

I get excited thinking about the next twenty years. What new breakthroughs will make unimaginable things possible? How much smaller and more powerful will computers become? Will we ever find the "best way" to use them in our schools? And, of course, what about Logo?

When I have a quiet moment to myself, I review a favorite fantasy about a time far in the future. A little boy comes up to me and says, "Grandaddy, Grandmomma says you remember when Logo first came out. Would you tell me what it was like back then?"

I smile, take the young lad in my arms, lean back in my chair, and begin...

FD 100! Tom

Conference Notes

On November 29, a seminar entitled "Logo and Mathematics" will be held at Oakland (MI) Schools from 4:30 to 8:15 p.m. The seminar will focus on how to use Logo to teach and enhance mathematics concepts in the classroom. The general session speaker will be Kathleen Martin, author of The Turtle's Sourcebook, coauthor of a popular monthly Logo column in The Computing Teacher magazine, and a cofounder of MicroQuests, a newsletter devoted to using Logo to explore mathematical ideas. Following a catered evening meal, two breakout sessions are scheduled, one for upper elementary and middle schools led by Martin, the other for high schools led by Lee Yunker. For more information, write Lynn Lieberman, Oakland Schools, 2100 Pontiac Lake Road, Pontiac, MI 48054, or call (313) 858-1896.

The Educational Computer Consortium of Ohio (ECCO) presents its fourth annual Educational Computer Fair on Friday and Saturday, October 12 and 13, in Cleveland, OH. There are several Logo presentations on the program, including "A Logo Curriculum for Grades K-6" by J. Kent Rawlings, "What Next? Beginning Logo in the Classroom" by Jane Toth and Donna Lanyi (Ed. note: Don't miss their TurtleTips column in each NLX issue.), "Logo: Fulfilling the Potential" by Thomas L. Beck, and "Using Logo in Secondary School Mathematics," by Michael T. Battista. Sharon Burrowes, author of several popular Logo articles (See April 1983 and January 1984 NLX), will speak on "Teacher Training - From Beginner to Programmer." For more information, write ECCO, 1123 SOM Center Road, Mayfield Heights, OH 44124, or call (216) 291-5525.

Tipps for Teachers

by Steve Tipps

Signs of Logo

The TEXTSCREEN in Logo is often overlooked as a place for important learning and exciting projects. One set of projects which highlights the possibilities and usefulness of the TEXTSCREEN is to make moving signs or bulletin boards.

You may want to work through this project skeleton for yourself, but it is not intended as a week of lesson plans. Instead, challenge students with the project of making a bulletin board on the computer like you see in front of banks or in Times Square. With this bigger idea in mind, some children will choose to work toward that goal.

Only PRINT and cursor commands are needed initially. All of the other ideas are intended as responses to problems and questions which almost inevitably arise. Children need variables and a WAIT. Some see a pattern and demand a better way. In this context, recursion makes a lot of sense. If children have already used recursion, then the project is a good extension.

MOVING WORDS

Making an animated sign or marquee or bulletin board is a simple project to start, but leads to variables, recursion, and testing. By moving the cursor around and PRINTing at different places on the screen, moving words are created.

The process is to PRINT a word somewhere on the screen, then erase it and draw it again somewhere else. Start with a simple HELLO or ALOHA. (MIT style commands are followed by LCSI ones.)

```
TO HELLO
PRINT1 "HELLO
END

TO HELLO
TYPE "HELLO
END
```

A cursor command puts the word on the tenth line down at the left side.

```
CURSOR 0 10 HELLO

SETCURSOR [0 10] HELLO
```

The apostrophes in MIT versions hold and print blank spaces. In the LCSI version, a backslash is used to print the very next character - in this case a space. If you do not have a backslash on the keyboard, try CTRL-Q.

After that bug is caught, you may find that the procedure works too fast for you to read. A delay would be good between the HELLO's. LCSI has a WAIT primitive, but you can write one for the MIT versions. (Hint: REPEAT 100 [])

PATTERN FOR HELLO

Moving HELLO across the screen becomes a very long procedure with the same commands used again and again for each change in position.

After you see how this works, you can move the HELLO across the screen from left to right by printing it in different horizontal positions.

```
TO MOVE.HELLO
CURSOR 0 10 HELLO
CURSOR 5 10 HELLO
CURSOR 10 10 HELLO
....
....
END

TO MOVE.HELLO
SETCURSOR [ 0 10 ] HELLO
SETCURSOR [ 5 10 ] HELLO
SETCURSOR [ 10 10 ] HELLO
....
....
END
```

The procedure looks as if it should work, but it doesn't. It just adds more of the same words on the line. You want one HELLO to move across the screen. Some method of erasing the old HELLO is needed. One way is to add CLEARTEXT between each HELLO. Or you could print a blank line with PRINT []. Another way would be to write an ERASE.HELLO procedure which blanks over the five spaces. Either METHOD works, but the latter is a little more precise.

```
TO ERASE.HELLO
PRINT1 ['     ']
END

TO ERASE.HELLO
TYPE [\ \ \ \ \ ]
END
```

Tipps continued

```
Move the cursor over and print HELLO
Move the cursor back and print blanks
Wait a little
Move the cursor over and print HELLO
Move the cursor back and print blanks
Wait a little
```

Recognition of this pattern should bring complaints. "Isn't there an easier way? The only thing that is changing is the first cursor number." The project has brought about the need for variables, recursion, and testing.

The variable could be called :SPACE, :HORIZONTAL, :OVER, or some other meaningful placeholder name. The procedure is simplified to three lines and a recursive call.

```
TO MOVE.HELLO2 :OVER
CURSOR :OVER 10 HELLO
REPEAT 100 []
CURSOR :OVER 10 ERASE.HELLO
MOVE.HELLO2 :OVER + 5
END
```

```
TO MOVE.HELLO2 :OVER
SETCURSOR LIST :OVER 10 HELLO
WAIT 10
SETCURSOR LIST :OVER 10 ERASE.HELLO
MOVE.HELLO2 :OVER + 5
END
```

Note that the LIST command is needed for the LCSI version instead of the square brackets. LIST puts the value of :OVER and 10 into a list for SETCURSOR to use.

Each recursive call moves the cursor over 5 to the right. This will cause a problem (and an error message) when the value of :OVER is greater than 39. A STOP test is needed to keep HELLO from running off the edge.

```
IF :OVER + 5 > 39  STOP
```

```
IF :OVER + 5 >  39 [STOP]
```

Try this line in different places in MOVE.HELLO2 to see where it works best. Also, you may need to play with the limits of the test to stop the sign before it falls onto the next line. When you have this all worked out, then you are ready to move HELLO from left to right.

The fun is not over yet. You can also move the sign from right to left. What would you need to change in the recursive line to change direction? The message can crawl or fly across the screen depending on the change in cursor values and WAIT. You could also make two cursor variables and move HELLO diagonally across the screen. You should also try the moving signs with PRINT instead of PRINT1 or TYPE. What is the difference?

STAR BRIGHT

Putting symbols on the TEXTSCREEN is another kind of display. For instance, an asterisk (*) makes a good star. The STAR procedure prints one star. REPEAT fills the whole screen with stars.

```
TO STAR
PRINT1 "*
END

REPEAT 1000 [STAR]
```

```
TO STAR
TYPE "*
END

REPEAT 1000 [STAR]
```

The sky is rarely filled with stars in a such regular way. Instead, they look rather random. You can also make them appear randomly on the screen. Use RANDOM as part of the input to cursor. The limits of the RANDOM are the size of the screen.

```
REPEAT 100 [ CURSOR RANDOM 40
             RANDOM 23 STAR]
```

```
REPEAT 100 [SETCURSOR LIST RANDOM 40
            RANDOM 23 STAR]
```

(Recall that LCSI needs LIST to put square brackets around the two numbers.)

You can control where you put the stars by changing the range on RANDOM. If you want stars only on the top part of the screen, you can set the second input to RANDOM 11. Putting stars in the upper left corner means changing both RANDOM numbers.

Putting stars in the upper right is a little more complicated. If you want the stars to be only in spaces 20 to 39, you have to move over 20 spaces before you choose the random number. Which one of these REPEAT statements do you think will give you the right numbers?

```
REPEAT 15 [PRINT 20 + RANDOM 20]
or
REPEAT 15 [PRINT RANDOM 20 + 20]
```

Tipps continued

Now you know how to "offset" the RANDOM numbers so that you can put stars, or anything, into any region of the TEXTSCREEN.

(Ed. note: This technique also works on the graphics screen. See this month's NLXual Challenge for a similar exercise.)

Stars can also be used to make constellation pictures. A piece of graph paper is handy for planning how to get the stars in the right place. Little stars can be made with periods.

ANNOUNCING

HELLO and STAR are simple models for the many projects students can do on the TEXTSCREEN. Any message can be displayed on the screen: birthday greetings, the lunch menu, special announcements, reminders, club meeting bulletins, etc. The computer becomes a class or school bulletin board. You could buy a software package, but students can write their own procedures personalized for their school or class.

Instead of HELLO, you can have a MESSAGE procedure which can be changed quickly by editing the procedure. You could even have the computer ask for the message.

```
TO SETUP
PRINT [WHAT IS TODAY'S MESSAGE?]
MAKE "MESSAGE REQUEST
END

TO SETUP
PRINT [WHAT IS TODAY'S MESSAGE?]
MAKE "MESSAGE READLIST
END
```

You could change SETUP so that it requests several different messages. Then, you could have several messages on the screen at one time, each one of them doing a different thing. One can move left to right, one right to left, and another flash randomly. Coordinating the messages is an aesthetic challenge.

Simple signs can be as exciting as complicated ones. One of most invigorating interactions I have had was with a group of primary LD students. I showed them a REPEAT statement which flashed my name on the screen 100 times. Within minutes, each of them had their own name PRINTed randomly on the screen. Adapt the messages and complexity to the interest of your students.

The TEXTSCREEN is a world unto itself with many more projects waiting to be developed.

(Ed. note: See also "Introduction to the Text Screen," by Glen Bull, NLX February 1983.)

Steve Tipps is the West Professor of Education at Midwestern State University in Wichita Falls, TX. He presents Logo workshops for school systems throughout the United States.

NLXionary

A Lectionary of Selected Logo Readings

by Griff Wigley

I bought a copy of the fall 1984 issue of Stuart Brand's latest venture, *The Whole Earth Software Review*, at the local newsstand. It is another Whole Earth hit. This particular issue contains several Logo articles, and is the source for this month's readings.

"Languages for Thinking about Learning: Eleven Logo's compared," by Jim McCauley

> A tidy summary of most of the currently available versions of Logo, interspersed with interesting comments from other computer folk. Art Kleiner, editor of Whole Earth, describes his personal experience of learning Logo just to edit this section of the review.

>Bob Albrecht has a short insert on why teachers should start with sprites when introducing Logo to kids. (I'd like to see a debate between Bob and Jim Muller on this issue!) Contributing editor Robert Scarola has a piece on "The Trouble with Logo" that describes three limitations to be aware of, particularly for advanced programmers.

>All this is in addition to Jim McCauley's short but concise dissection of the various Logo's, including his clear preferences.

NLXionary continued

"Baby Logo's" by Pamela Sharp, with an introduction by Robert Scarola

> Three Logolike turtle graphics packages are reviewed: Turtle Toyland Jr for the Commodore 64, Cyberlogo Turtle for the Apple, and Turtle Power for the IBM.

>Sharp prefers Cyberlogo and Turtle Power because they are essentially Logo without the word and list processing capabilities, and with some added features to ease the transition to the full language. Toyland, on the other hand, is joystick controlled, and its procedures cannot be edited.

>This whole issue of easy and/or instant Logo's is another area for debate and discussion. What are the longer range benefits and problems if they are used? (Ed. note: See also "The Issue of Instant," by Steve Tipps, NLX December 1983.)

"Using Logo, Making Science," by Robert Scarola

>Scarola, an elementary school teacher, describes his attempts to use Logo to develop a curriculum project for teaching science. Exploring patterns in nature through the geometric patterns and mathematical beauty in Logo's turtle was his focus.

>"I discovered we could move from the broad perspective of galaxies and solar systems to the available world of landscapes, rocks, minerals, plants, animals, and machines, and end close to home with the body, nutrition, reproduction, bones, blood, and cells." Pairing field trips with assignments in Logo, Scarola describes a year long process that makes me want to move to Bolinas, California, to enroll my kids in his school.

>He also adds a plug for instant Logo's, arguing that they get kids into thinking about patterns at an earlier age.

>This article is worth the price of the review alone for all you teachers out there on the front lines.

Griff Wigley is a co-owner of Family Computing, Inc., a facilitator in the Faribault (MN) Public Schools, and a school board member of Prairie Creek Community School.

TurtleTips

by

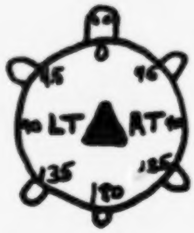
Donna Lanyi
and Jane Toth

Now that the school year has started, you are probably looking for activities to motivate and help students learn more about Logo. After working on some precomputer activities and experimenting with "turtling around," your students might be ready for planning Logo projects and then producing them on the computer. Here are some ideas.

TEACHER HINTS

A roll of plain old masking tape is an aid we have found useful for students planning simple pictures. They tape out their design on the floor, walk through the design, and list the commands needed. This process allows students to visualize the design before working at the computer. It also encourages a group process, because the figure on the floor is public.

If your students have difficulty understanding the degrees used in turning the turtle, try the "paper plate technique." Have students draw a turtle on a small paper plate. Add LT and RT (if appropriate) and selected numbers for the turns. The plate turtle shown here has numbers for turns of 45, 90, 135, and 180.



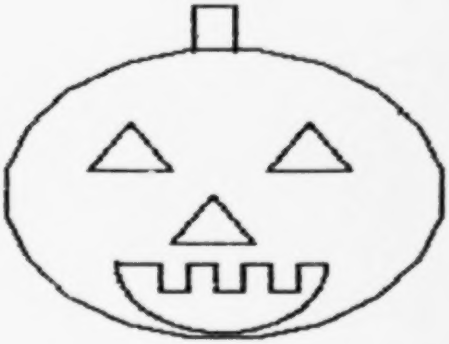
We have successfully used the plate turtle by having students place it on the computer screen aligned with the Logo turtle. Then they match the direction they want to turn with the numbers on the plate. As students master the concept, we add more and more numbers to the plate.

Additional embellishments include decorating the plate, naming the turtle, and the like. (Ed. note: See also "Logo Tips from Punahou," Electronic Learning, April 1983, page 73, for more related ideas.)

TurtleTips continued

PROCEDURE OF THE MONTH

Since it is October, our procedure this month coincides with the Halloween theme. Ask your students to match our Jack-O-Lantern design or create one of their own. Encourage students to draw other designs on the Halloween theme.



If you are interested in a free listing of this procedure, please send us a self addressed stamped envelope.

We invite elementary teachers and students to send their procedures or ideas to be shared. Donna Lanyi and Jane Toth, Beall Avenue School, 716 Beall Avenue, Wooster, OH 44691.

Donna Lanyi and Jane Toth use Logo in their elementary classrooms in the Wooster (OH) City Schools.

Logo Research in Progress

Researchers at the University of Maryland's Center for Young Children are involved in investigating the use of microcomputers with 3, 4, and 5 year old children. A robot turtle, Instant Logo, and Delta Drawing have been used with various activities. A special dimension of the project is parent involvement. Parent workshops, parent volunteers in the project, and sessions with both parent and child provide open communication and exchange of ideas. For more information, contact June Wright, Computer Discovery Project Coordinator, 1229 HRW Benjamin Building, University of Maryland, College Park, MD 20742, or call (301) 454-2341.

Listful Thinking

by

Glen Bull and Paula Cochran

Pick-a-Treat

Witches and pumpkins and monsters, oh my! It will soon be that time again, when students are so busy thinking about ghosts and goblins that getting them to concentrate on school work can be difficult.

We have used Halloween words and images this month in hopes that Logo and list processing will help you take advantage of student interests in ghouls to work on some language arts goals. If you teach secondary students, the underlying activities can be used with them, too, perhaps without the Halloween emphasis.

TEACHING ENGLISH OR TEACHING LOGO

Some teachers have expressed concern that too much of their instructional time goes to teaching Logo and not enough to their content area when they attempt to use Logo activities in class. We suggest that, in many cases, this can be avoided if teachers provide key Logo procedures for accomplishing a particular content-related goal. Although some students may go on to dissect such procedures in a programming class, for example, they need not understand the details for the moment.

Last month, we tried to provide a model of this approach to using Logo in the classroom. We took on the guise of an English teacher to explore some aspects of poetry using such Logo tools as a procedure called PICK (see Listful Thinking, NLX, September 1984.)

This month we continue to wear this mask while suggesting some other ways to make lists and use PICK. As you recall, PICK randomly selects a word from a list. This is what PICK looks like and how you use it:

```
TO PICK :LIST
OUTPUT ITEM 1 + (RANDOM COUNT :LIST) :LIST
END
```

```
PRINT PICK [GHOST WITCH CAT SKELETON]
SKELETON
```

Try PICK with your own list of words. If you have trouble, or are using Terrapin Version 1 or Krell Logo for the Apple, check last month's column for hints and translations.

Listful Thinking continued

BORING OR BRILLIANT

One frequently heard complaint from language arts teachers is that children use boring adjectives. Kathy Lascala, an LD teacher we know, calls them "the terrible ten" - nice, okay, good, etc. Halloween is an excellent excuse for talking about more expressive description words. The class might come up with a list such as the following, which we have put into a procedure using PICK.

```
TO ADJECTIVE
OUTPUT PICK [SCARY WILD MYSTERIOUS SPOOKY]
END

PRINT ADJECTIVE
SCARY
```

METAPHORICALLY SPEAKING

With a couple of more lists of Halloween words generated by the students, you could work together on writing some metaphors about Halloween. A metaphor is an implied comparison in which comparative words such as "like" and "as" are not used. The procedure below yields some delightful metaphors:

```
TO METAPHOR.POEM
PRINT SE [HALLOWEEN IS] ADJECTIVE
PRINT SE [HALLOWEEN IS A] NOUN
PRINT PLACE
END

TO NOUN
OP PICK [GHOST GREMLIN MONSTER WITCH]
END

TO PLACE
OP PICK [[IN THE WOODS] [UNDER MY BED] [IN
   THE MOONLIGHT] [IN THE CLOSET]]
END
```

When you call METAPHOR.POEM here's what you might get:

```
METAPHOR.POEM

HALLOWEEN IS SPOOKY
HALLOWEEN IS A GREMLIN
UNDER MY BED
```

or perhaps

```
HALLOWEEN IS MYSTERIOUS
HALLOWEEN IS A WITCH
IN THE MOONLIGHT
```

If this has been a class project, the students could vote on the verses they like best and make copies to illustrate or share with other classes.

BIG, BRIGHT, AND BOO-TIFUL LISTS

Using PICK to choose from lists which follow it within brackets is fine for short lists, or lists you only want to use one way. Many times, however, it would be helpful to have a name for the list we typed. We cannot, for example, ask Logo to print out the list of adjectives we used above, because the list itself has no name.

Besides putting lists into procedures, as in NOUN, ADJECTIVE, and PLACE, you can make Logo lists as follows:

```
MAKE "TREATS [CANDY POPCORN COOKIES]
```

If we want to know what the treats are, we ask for the list or THING associated with the name:

```
PRINT THING "TREATS
CANDY POPCORN COOKIES
```

There is a short-hand way of typing the THING request, which makes use of the colon (or "dots") :

```
PRINT :TREATS
CANDY POPCORN COOKIES
```

Of course, we can PICK a treat:

```
PRINT PICK :TREATS
POPCORN
```

Now we can make a long list of words which we can print or use in several ways. Since we're doing poetry at the moment, this is how such a list could be used to play with alliteration:

```
TO B.WORDS
MAKE "B.LIST [BIG BEASTLY BAD BOOTIFUL]
END

TO ALLITERATION.POEM
PRINT [ON HALLOWEEN THE]
REPEAT 2 [PRINT PICK :B.LIST]
PRINT SE NOUN [WAS]
PRINT PICK :B.LIST
END

B.WORDS
ALLITERATION.POEM
```

Listful Thinking continued

ON HALLOWEEN THE
BIG
BAD
WITCH WAS
BEASTLY

** Note ** When you create a list within a procedure, as in B.WORDS above, you must call the procedure in order to give Logo a chance to MAKE the list. If you change the contents of the list by editing the procedure, don't forget to run the procedure in order to MAKE the list again, with the changes. If this seems confusing, think of a SQUARE procedure. Just writing the procedure doesn't cause the square to be drawn. Calling it does.

We hope that you and your students have fun using Logo to talk about ghosts and goblins, and maybe even about some poetry concepts such as metaphors and alliteration. We also hope that this way of using Logo to implement content-specific activities becomes less mysterious and even more exciting than Halloween!

Glen Bull is a professor in the University of Virginia's School of Education, and teaches Logo courses at both the graduate and undergraduate level. Paula Cochran is a reformed English major who studied linguistics at the University of Cambridge. She is now a speech-language pathologist working with language-disabled children.

前 100

by
Hillel
Weintraub

In this month's column, I'd like to tell you a little about the growth and direction of Logo in Japan.

In order to understand just how far Logo has come in two years, it is important to see what the state of Logo was in October, 1982. At that time, I was looking for information about Logo. I was directed to Terrapin's Japanese representative, the BYNAS division of the UNY company, a forward-looking company which had decided that Logo had something important to offer Japanese education.

When I visited the BYNAS office and looked around, I discovered that there were no Japanese publications on Logo and there were no Japanese computers running Logo.

However, two key events were in progress: the Japanese translation and publication of Mindstorms was being finalized, and Professor Papert's first visit to Japan was being arranged. These two events set the Japanese Logo world afire.

The present state of Logo in Japan, less than two years later, is this: there are now some 13 different Logo publications in Japanese on the shelves of the BYNAS office; every major computer company in Japan - NEC, Sony, Fujitsu, Sharp, Hitachi, and Epson - all have Logo versions running.

Two of these, Sony and Fujitsu, have bundled Logo with their latest models. At least three companies are actively working to create Logo languages for other computers and other models. There are two Logo-only publications, The Logo Journal, publishjed by UNY Company, and Logosome Z's, published by Geodesic. There is also an active Logo research / study group.

Perhaps the most important development is potential rather than actual as of this date: that is, the bundling of Logo with the MSX computers. These computers, marketed by nearly every computer company in Japan, all have the same operating system and can run virtually the same software. At present, most MSX software (99%) is of the game variety.

The MSX version of Logo has been completed by LCSI, and negotiations are underway to have it bundled with each MSX machine. What this would do to the place of Logo in Japan is mindboggling. Educators and others interested in Logo are considering the educational implications.

In future columns, we'll consider some of these implications, too, as well as some of the people and publications instrumental in Logo's Japanese development. I will also share with you some of the differences between the structure of the educational system in Japan and the USA and how this will affect Logo use here.

Until then, MAE 100!

Hillel Weintraub is a teacher at Doshisha International High School in Kyoyo-fu, Japan, and is the president of the Society for Microcomputing in Life and Education (SMILE).

Teacher to Teacher

Triscuits and Task Cards: Aids for Elementary Grades

Reviewed by Regina Bowden and
Anne Cairns Federlein

Crackers come in a variety of shapes and sizes. They serve a variety of people for a variety of reasons. It is comforting to have some on hand at all times. Despite their similarities, preferences revolve around differences. That seems to be the case with many things in life besides crackers. Logo task cards are like that, too.

This review of five sets of Logo task cards is written to help you discriminate. Each classroom has unique needs. Just as crackers in the pantry provide a measure of comfort, task cards alongside the computer can help enrich the user, whether child or adult.

Included are Logo Task Cards, from Computer Skill Builders, Logocards: Activities for Apple Logo and Logocards: Activities for TI Logo, from Scott, Foresman, Exploring Logo: Apple, and Exploring Logo: Commodore, from Sunburst Communications, One Key Logo and Logo, from Lincoln (NE) Public Schools, and Logo Premiers Pas avec la Tortue, from Micro-Ami, Inc.

SIMILARITIES

* Authors agree that task cards are supplementary to the Logo curriculum. They provide ideas for teachers, parents, and students, when new ideas run low. (Haven't we all had one or more of those days?)

* All five sets of cards feature turtle graphics.

* All are arranged by categories, with simple tasks addressed first. Most cards can be used effectively out of order.

* All sets make excellent resources for teachers who need new ideas, lesson planning, and clarification of concepts.

* Authors mention that use by a small group of two or three students is preferred over individual use. There is more language exchange and discussion about the problem when two or three students work together. This leads to a deeper understanding of the principles involved.

DIFFERENCES

Logo Task Cards
Authors: Andy Howard
Roger Goodberlet

Logo versions supported: Apple Logo, Terrapin Logo, and Krell Logo for the Apple computer; IBM Logo, Dr. Logo, and Waterloo Logo for the IBM PC; Atari Logo

Ten instruction cards give a brief but thorough introduction to the Logo philosophy. A list of skills is included, not to be used as a strict set of objectives, but as a helpful guide to those who must "justify" working with Logo in the classroom. The skills list also helps teachers with little Logo experience to identify new skills learned by the students.

The activity cards give an extended view of the power of Logo, and include math and word and list operations as well as turtle graphics. A response booklet is included which helps the student get into the habit of keeping a journal of experiences for reference. A benefit of owning task cards is the file of creative suggestions for challenging students. This set has some interesting challenges using RANDOM, a honeycomb pattern, and a quilt. These challenges keep that very active student busy and learning!

There is also a set of 60 visual masters available to be used independently or in conjunction with other materials. The cards can be used independently by children ages 9 and up.

Exploring Logo: Apple
Exploring Logo: Commodore
Authors: Karl Zinn
Gordon Leacock

Logo versions supported: separate decks of cards for Terrapin Logo and Krell Logo, Apple Logo, and Commodore Logo

Of the five sets of cards, this set contains the most technical information. Geared for older students (over 12 years) and adults, there is an activity, exploration, and explanation section on each card. The explanation section also contains a "quick review". Some cards have a "powerful idea" and a "useful technique" area for tips about the concepts covered on the card.

In addition to turtle graphics, there are cards which address words and lists, music, mathematics, and sprites (Commodore version only). There is a quick reference guide to Logo commands and conventions, and a glossary.

Teacher to Teacher continued

The 50 page guidebook contains suggestions for creating the classroom environment, Logo philosophy, and applications. The accompanying student diskette contains valuable sample programs and utilities. In addition to being an aid for learning the Logo language, this set of cards could be used as a quick reference to commands and operations.

Logocards: Activities for Apple Logo
Logocards: Activities for TI Logo
Authors: Elaine Blitman
Barbara Jamile

Logo versions supported: Apple Logo for the Apple computer, TI Logo for the Texas Instruments 99/4 and 99/4A computers

Each set contains 100 5x8 cards, arranged according to the commands and operations presented. A typical card contains a particular topic, an example to try (individually or in a small group), a question to answer, and more samples for experimentation. Challenge cards mark the end of each group of concepts and reinforce what was learned.

Children as young as 8 years old could work with the cards alone or in groups of two or three. However, the authors suggest that even 6 year olds are capable of using the cards.

The text on the cards is in primary print, easy to read by the young child, with the keys to be used emphasized in darker print. They are brightly color coded and printed on durable heavyweight paper. If a teacher laminates these cards, then the children could use the backs to record their comments in a log.

Teachers and children alike would be comfortable using this tool!

One Key Logo (two sets)
Logo (one set)
Authors: Chuck Friesen
Kathy Parker
Jerry Shedlosky

Logo versions supported: Terrapin Logo and Krell Logo for the Apple computer, Commodore Logo for the Commodore 64

These three sets of cards are quite inexpensive and have been classoom tested by 350 teachers in the Lincoln Public Schools in Nebraska. The first set of One Key Logo is meant for use by "non-reading" students. Clear simple drawings lead the child through exploration on the computer using the single key procedures on the accompanying diskette.

Some nice features of this easy to use version include the ability to draw circles of varying sizes for use in creative drawings, to undo, to erase, to name procedures and recall them easily, and, of course, to save drawings.

The second One Key Logo set contains written instructions. Children with minimal reading skills who need some room to stretch on their own could handle these cards. This is the perfect opportunity for a child who may not normally be a star to shine as the leader of a small group involved in Logo activities. Some classrooms use older student helpers. These cards could help them work wtih small groups of younger students.

The Logo set of cards is for older children and adults, and lead them through an introduction to the Logo language. The focus is on exploration followed by creative discovery. Parent helpers would also benefit from practice with these cards!

(And, don't forget, some adults prefer to practice first in private. Providing a quiet room for adult exploration at scheduled times is thoughtful, and would help boost the self confidence of the your adult volunteers.)

Logo Premiers Pas avec la Tortue
Author: Marie De'carie

Logo version supported: French Apple Logo (LCSI) for the Apple computer

Exploring these cards was lots of fun! It has been quite a while since college French, and we cajoled the younger set into coming to our rescue. With the aid of students in a high school French class, we finally got going!

Once again, the emphasis in these cards is on exploration, explanation, and encouragement to play and discover. The major difference is that they are written in French! This brings to mind a strength of the Logo language: the ability for users to define their own commands.

These cards were written to accompany the French version of Logo, available for the Apple from LCSI. With a bit of creativity, however, they could be put to good use with any version of Logo by a French teacher or student. Both elementary and high school students might enjoy using the cards as part of a unit on France. The cards are written for fourth through sixth grades.

Teacher to Teacher continued

IN SUMMARY

As educators concerned wtih the danger of too many drill programs in the classroom and not enough experience with problem solving, we found that the authors of these materials are very sensitive to this situation. The cards focus on presentation, exploration, and discovery learning. Task cards can provide that spark of an idea for creativity to grow.

For a new Logo user, this complex language can be quite overwhelming, if viewed holistically. Task cards allow users to isolate individual concepts and build on them slowly.

There is also a population of adult learners who enjoy the computer and the challenges it provides. Task cards can be a non-threatening way to improve reading and thinking in the privacy of their homes.

Parents who purchased the Logo language and need suggestions on implementation will find task cards very meaningful.

Logo task Cards, available from Computer Skill Builders, 3130 North Dodge Boulevard, PO Box 42050, Tucson, AZ 85773. (602) 323-7500. $14.95 for 60 cards and student booklet. Extra student booklets cost $7.50 for a set of 10. A set of visual masters costs $19.95.

Logocards: Activities for Apple Logo and Logocards: Activities for TI Logo, available from Scott, Foresman Electronic Publishing, 1900 East Lake Avenue, Glenview, IL 60025. Each set contains 100 activity cards and a user's guide for $19.95.

Exploring Logo: Apple and Exploring Logo: Commodore, available from Sunburst Communications, Pleasantville, NY 10570. (914) 769-5030. $49.00 for a set of 26 cards, 62 page guidebook, and a student diskette.

One Key Logo (2 sets) and Logo (1 set), available from Lincoln Public Schools, Educational Technology, 720 South 22nd Street, Box 85889, Lincoln, NE 68502. $5.00 per set of cards, $3.00 per diskette. What a bargain!

Logo Premiers Pas avec la Tortue, available from Micro-Ami, Inc., 16 rue Georges, Ormstown, Quebec JOS 1KO Canada. $19.95 for a set of 37 cards.

For comments, questions, and concerns on the task cards, please contact Logo Information For Teachers (LIFT) at PO Box 5396, Plymouth, MI 48170.

Avance 100!

Regina Bowden is the Director of Education at Computer Horizons Training Center in Livonia, MI. Anne Cairns Federlein is a professor in the Early Childhood Area, School of Human and Educational Services, Oakland University, Rochester, MI.

Logo Disserts:

Dissertations Dealing with Logo

by Barbara Elias

Can Logo programming facilitate the development of problem solving ability? An investigation of this question was conducted using Logo programming tasks.

Problem solving was defined as the accomplishment of a goal which is not easily attainable. Problem solving requires action or activity in order to reach a desired goal. Programming, an activity analagous to problem solving, was selected for this study because it provides a rich problem solving medium with varied possibilities.

Since the Logo language is relatively easy for children to use, it was selected for use in the study.

A modification of the Polya model was used to relate the essential problem solving / programming tasks.

1. Defining the problem
2. Devising a plan
3. Gathering information
4. Executing the plan
5. Revising the plan
6. Evaluating the results

48 fourth grade children spent a few hours a week for two and one half semesters on Logo tasks. Instruction included concept development of procedure definition, editing, recursion, inputs, conditionals, and variables. When a child could use a concept successfully in similar and diverse cases, mastery was acheived.

Logo Disserts continued

In general, the children showed gains in recursion and communication tasks, but not in problem solving tasks.

Factors which affected concept development included time available, reading level, arithmetic and typing ability, design drawing skill, and ability to attend to the tasks.

An additional result of this study is a set of hypotheses on programming concepts, heuristics, transfer, and attitude. In a real sense, these same hypotheses continue to be of interest to researchers today.

>>>Statz, J. A. (1974). The Development of Computer Programming Concepts and Problem Solving Abilities among Ten Year Olds Learning Logo (Doctoral dissertation, Syracuse University, 1973). Dissertation Abstracts International, <u>74</u>, 5418-B.

(Ed. note: Each month, Barbara Elias highlights a dissertation or thesis dealing with Logo. She is accepting copies of recent research results for the NLX Dissertation and Thesis Repository. Reports may be mailed to her in care of NLX, PO Box 5341, Charlottesville, VA 22905.)

Barbara Elias is an assistant professor in the Education Department of Virginia State University in Petersburg, VA, and a doctoral candidate at the University of Virginia.

Logo Featured at NC Conference

The North Carolina Educational Microcomputer Conference, held in Winston-Salem, NC, September 20-21, featured four sessions dealing with Logo. Sheila Cory suggested a cautious approach to implementing Logo in the elementary schools, and gave suggestions on curriculum development and program evaluation. Mary Jenkins presented an introduction to Logo to those with no previous experience. Judy LeCroy discussed how to stretch five computers to cover the needs of over 700 students, and how to train Logo teachers. Elaine Bologna shared information on how to integrate Logo effectively into a traditional curriculum, including ideas for individual students, small groups, and the whole class. For additional conference information, contact Jeanette Gann, NC Regional Educational Center, Region 5, PO Box 21899, Greensboro, NC 27420, or call (919) 379-5764.

Q and A

by
Jim McCauley

Q. I'm just starting out with list processing, and I can't sort out all the differences among SENTENCE, LIST, FPUT, and LPUT. They are all used to make lists; why are there so many, and what are they used for?

<[-?-]>

A. These primitives (which I call "list constructors") caused me a lot of befuddlement when I started using them, too. Each of the four has its special properties and applications.

<u>SENTENCE</u>

The list constructor that nearly everyone starts with is SENTENCE. It is very simple; ordinarily, it takes two objects (words or lists) and simply mashes them together into one list.

```
SHOW SENTENCE "A "B
SHOW SENTENCE "A [B]
SHOW SENTENCE [A] "B
SHOW SENTENCE [A] [B]
```

all return [A B] as a result. As you can see, SENTENCE removes the distinction between words and lists when it smashes them together. For certain applications, it is handy to do this. Here's an example:

```
TO GREET :NAME
PRINT SENTENCE "HI, :NAME
END
```

This procedure works properly no matter whether the input is a word or list because SENTENCE "ignores" the first level (the outermost) of brackets in :NAME if :NAME is a list.

```
GREET "DOROTHY
HI, DOROTHY

GREET [WIZARD OF OZZZ]
HI, WIZARD OF OZZZ
```

<u>LIST</u>

LIST, on the other hand, is very particular. It makes a list of its two inputs and carefully preserves the individual nature of each.

Q and A continued

```
     SHOW LIST "A "B gives [A B]
     SHOW LIST "A [B] gives [A [B]]
     SHOW LIST [A] "B gives [[A] B]
     SHOW LIST [A] [B] gives [[A][B]]
```

SENTENCE and LIST accept more than two inputs if the primitive and its arguments are surrounded by parentheses.

```
     SHOW SENTENCE ("I [AM A][LOGO LIST])
     [I AM A LOGO LIST]
```

In most Logo versions, SENTENCE and LIST accept one argument if both the primitive and its argument are surrounded by parentheses.

```
     SHOW (LIST "FRED) gives [FRED]
```

This may seem pointless, but sometimes it is handy to be able to turn a single word into a list. For example, if you have defined a series of procedures whose names are letters of the alphabet (such as in Instant Logo),

```
     RUN (LIST READCHAR)
```

is a handy instruction to use.

THE MAVERICKS

FPUT and LPUT are a breed apart. FPUT (First PUT) takes two arguments: an object (either a word or a list) and a list. It returns a new list composed of the object as the first element of the new list and the second argument as the rest of the new list.

```
     SHOW FPUT "A [B] gives [A B]
     SHOW FPUT [A] [B] gives [[A][B]]
     SHOW FPUT "A [[B]] gives [A [B]]
     SHOW FPUT [A] [[B]] gives [[A][B]]
```

LPUT (Last PUT) has a similar effect, except that it is used to place a new object at the end of a list.

FPUT and LPUT are used regularly in recursive processing of lists, especially in procedures of the type I call "scanners." These are procedures that recursively decompose lists, then automatically reconstruct them after treating each element. LPUT often puts in an appearance in procedures that reverse words or lists. Here is a typical application of FPUT's properties.

```
     TO ANTI :LIST.IN
     IF EMPTYP :LIST.IN [OUTPUT []]
     OUTPUT  FPUT ANTI.IT  FIRST  :LIST.IN
ANTI BUTFIRST :LIST.IN
     END

     TO ANTI.IT :OBJECT
     IF  WORDP :OBJECT [OUTPUT WORD  "ANTI
:OBJECT]
     OUTPUT FPUT WORD "ANTI FIRST  :OBJECT
BUTFIRST :OBJECT
     END
```

Try this out in any version of LCSI Logo and see if you can figure out the two different kinds of things that FPUT is doing in each of these procedures. The same code works in MIT Logo if you eliminate the outer brackets in the IF lines. WORD? and EMPTY? primitives are also needed.

A clear understanding of the list constructors is vital to the development of list processing skills. I encourage all readers to experiment with them, using all imaginable combinations of inputs. That's the best way to learn how they work and how they can be used. Good luck!

(Ed. note: If you have questions about Logo programming techniques or the thinking which leads to Logo programming, send them to NLX Q & A, PO Box 5341, Charlottesville, VA 22905. Each month, Jim McCauley answers selected questions in this column.)

Jim McCauley is a Coordinator of Computer Education for the Santa Clara (CA) County Office of Education, and has written Logo articles for many national publications.

Logo Notes

Al's Logo WIZ-KIT is a Logo utility diskette which contains procedures to print Logo pictures, manage memory, and more. It supports both MIT and LCSI Logo versions for the Apple, and works with many different printers and interface cards. It is available for $19.95. For a free brochure, write Hands-On Training Company, 4021 Allen School Road, Bonita, CA 92002.

A videotape entitled "Apple Logo - The Computer Language Tht Holds Your Hand" is available from COMPUVID, INC. The tape includes both turtle graphics and list processing. For more information, contact COMPUVID, Inc., 815 North Royal Street, Alexandria, VA 22314, or call (703) 683-3234.

More Logo Notes

Computer Outlet offers DR Logo for $69 and Atari Logo for $79. Computer Outlet, Park Place, Upper Level, 1095 E. Twain, Las Vegas, NV 89109. (800) 634-6766.

Have you been looking for a screen dump you can use within Apple Logo? CALC-U-WARE has such a program available for the Apple Imagewriter printer. Called L.I.P. (for Logo Imagewriter Patch), it prints the contents of the graphics screen directly when the command IMAGE is typed. $19.95 from CALC-U-WARE, 32921 1st Avenue South, Federal Way, WA. (206) 874-2540.

A free catalog describing Logo teaching materials exclusively is yours upon request. Send a postcard to K-12 MicroMedia, Dept. L, 172 Broadway, Woodcliff Lake, NJ 07675. The 12 page catalog includes reference and resource books, textbooks, activity cards, workbooks, and other Logo related materials. Logo versions for the Apple, Atari, Commodore 64 and IBM PC are supported. Items may be ordered on a 30-day on-approval basis.

Krell Software Corp. announces its new improved MIT Logo for the Apple, available October 1, 1984. New features include upper and lower case letters, faster loading, recovery of accidentally deleted files, and improved documentation. The price is $89.95. For more information, contact Krell Software Corp., 1320 Stony Brook Road, Stony Brook, NY 11790, or call (800) 245-7355.

Lloyd Rieber has completed his masters thesis entitled, "The Effect of Logo on Increasing Systematic and Procedural Thinking According to Piaget's Theory of Intellectual Development and on its Ability to Teach Geometry Concepts to Young Children." To obtain a copy, send $4.00 to Lloyd Rieber, PO Box 305, Bluewater, NM 87005.

Are you interested in learning more about Piaget's theory to support your interest in Logo? If so, you may want to pick up a copy of The Piaget Primer: Thinking, Learning, and Teaching, by Ed Labinowicz (Addison Wesley, 1980). His most recent book is Listening to Children: New Beginnings for Teaching, which highlights children's activities as they invent their own computational algorithms, taking control of their learning.

NLXual Challenges

by
Robs Muir

Random Rings and Things

The RANDOM command can be a valuable and unpredictable tool in the Logo environment. Although the RANDOM command is not truly a random primitive, this pseudo-random number generator has a rich potential for Logo experimentation.

PRINT RANDOM :NUMBER will display a random number from 0 up to the THING (associated with) :NUMBER. For example, PRINT RANDOM 10 will print a number from 0 through 9. This construction can also be used with other primitives.

Some examples include REPEAT 100 [PRINT RANDOM 100], FD RANDOM 100, RT RANDOM 180, PRINT ((2 * :PI) * RANDOM 20), OUTPUT (RANDOM 10) + (RANDOM 10), SETH RANDOM 360.

Here is a group of procedures which will draw circles randomly on your graphics screen.

```
TO DRAW.CIRCLE
RPEAT 36 [FD 1 RT 10]
END

TO RANDOM.CIRCLE
; [MIT LOGO VERSION]
PENUP
SETXY (RANDOM 140) (RANDOM 120)
PENDOWN
DRAW.CIRCLE
END
```

For LCSI Logo, use SETPOS LIST (RANDOM 140) (RANDOM 120) in place of the SETXY line in RANDOM.CIRCLE.

MINOR REPAIRS

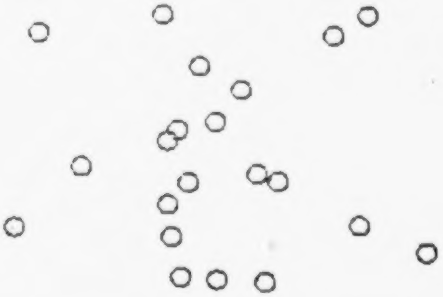
A disadvantage with these procedures (as you will discover when you try REPEAT 100 [RANDOM.CIRCLE]) is that the circles are all drawn in the upper right quadrant of the screen. How would you change the RANDOM.CIRCLE procedure so that circles are drawn all over the screen? One suggested method appears at the end of the column.

These procedures could become important parts of interesting games and other activities, if not for the fact that, when used repeatedly, our random number generator sometimes picks the same numbers.

NLXual Challenges continued

HERE COMES THE CURVE BALL

This month's problem is contributed by Jorge Luis Mendivelzua of Argentina. Jorge wished to write a Logo game for his daughter which would generate 20 random "targets" similar to the circles in the procedures above. The difficulty came when he noticed that he sometimes got duplicate circles.



Can you take Jorge's problem and write a procedure that will generate a list of 20 random numbers between 0 and 100 in which there is no duplication? For consistency of output, have your computer print this list on the screen.

If you would like to go even further, use the idea from your random number list to figure out how to draw 20 circles at random locations on the screen so that none of them overlap or intersect.

When you have a working solution (or even a valiant attempt), send your solution, along with a self addressed stamped envelope to:

NLXual Challenges
Attn: Robs Muir
1688 Denver Avenue
Claremont, CA 91711

We will compile the contributed solutions, select the most interesting or unusual ones, and redistribute them to each contributor.

Our "challenge community" already involves NLX readers in six countries, so the contributions are quite diverse. Solutions are available for each problem, but you must contribute your attempted solution (however meager) to receive these "answers".

>By the way, to fix RANDOM.CIRCLE, change the two RANDOM commands to (140 - RANDOM 280) (120 - RANDOM 240).

Robs Muir is a teacher in the Claremont CA Unified School District and Claremont Graduate School. He presents Logo workshops in the southern California area.

Logo Teacher Trainer Network Formed

Logo Information For Teachers (LIFT) is forming a national network of groups and individuals who are involved with Logo teacher training and Logo product development. The group will explore ways to exchange information, offer professional support, and identify and help to solve common problems in teacher training and product development. If you train Logo teachers or create Logo products, this is the group to join. Private business firms as well as school affiliated individuals and organizations are welcome. Write to LIFT, PO Box 5396, Plymouth, MI 48170, or call Regina Bowden at (313) 464-8088.

The National Logo Exchange, copyright (c) 1984 by Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly, September through May, $25 (US) per year, mailed first class from Charlottesville, VA. $5 (US) additional per year for addresses outside the USA, Canada, and Mexico. The opinions expressed by the authors are not necessarily those of The National Logo Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress Street, Salem, MA 01971. Address other correspondence to: The National Logo Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor..Tom Lough

THE NATIONAL LOGO EXCHANGE

Volume 3 Number 3 FORWARD 100! November 1984

Logo Trail of Tears

by C. M. Chmielowski

The little boy's sobbing devastated me. He and 23 other second graders had ridden the bus for half an hour to reach the school district's computer lab. And Logo had frustrated him.

The children and their teacher had only two or three hours in my Apple lab. Their brief introduction to Logo must convince them of Logo's value and must provide the tools they need to continue working with the language when they return to their classroom.

In that short amount of time, I must:

1. introduce Logo to all the students and their teacher.
2. work with more than one learner at a time.
3. insure productive and challenging explorations with Logo, not frustrating ones.

Time constraints dictate that the children learn only Logo primitives here in the lab instead of using specially written "Instant" programs. Otherwise, teachers and young children returning to the classroom are frequently confused about how to use the Logo language available at their school.

NO MORE TEARS

Here is a program I developed to help teach the turtle graphic primitives in a way that is not frustrating to young children. It is challenging for first and second graders - more so than an "Instant" program. However, it does not require typing numbers, so its demands are within a 6 or 7 year old's grasp. Children learn to spell the command, and realize they must press <return> after each command. This program can serve as a transition between a kindergartner's use of an "Instant" program and regular Logo. (Note: The space between [D] and the final parenthesis in Terrapin Logo is critical.)

```
    TO HELP
    MAKE "ANS REQUEST
    IF (ANYOF :ANS = [FD] :ANS = [BK] :ANS
= [RT] :ANS = [LT] :ANS = [D] ) CHECK  :ANS
    PRINT [TYPE FD, BK, RT, LT, OR D]
    HELP
    END

    TO CHECK :ANS
    IF :ANS = [FD] FD 10
    IF :ANS = [BK] BK 10
    IF :ANS = [RT] RT 30
    IF :ANS = [LT] LT 30
    IF :ANS = [D] DRAW
    END
```

HERE COME THE NUMBERS

When students are comfortable with these turtle commands, they are ready to include numbers with them. The following program permits this, with a turtle that moves slowly enough that children can follow its movements easily regardless of the input numbers.

Its inspiration was Tom Lough's "Slow Turtle" in the September 1983 NLX. The inclusion of a check for numbers larger than 100 prevents accidentally typing a large number. It also discourages endless "wrapping around," thus encouraging more productive experimentation with Logo. When a student is ready to use regular Logo, a simple Ctrl-G is all that is needed. (Note: The space between "LT and the right parenthesis is critical.)

```
    TO HELP1
    MAKE "ANS REQUEST
    IF :ANS = [DRAW] DRAW
    IF :ANS = [] MAKE "ANS [[]]
    IF ALLOF (ANYOF FIRST :ANS = "FD FIRST
:ANS  = "BK FIRST :ANS = "RT FIRST  :ANS  =
"LT ) (NUMBER? LAST :ANS) CHECK1 :ANS
    PRINT [TYPE A COMMAND,  A SPACE, AND A
NUMBER.]
    HELP1
    END
```

Tears continued

```
     TO CHECK1 :ANS
     IF  LAST :ANS > 100 PRINT [PLEASE  USE
NUMBERS SMALLER THAN 100.] STOP
     IF  FIRST :ANS = "FD  REPEAT LAST :ANS
[FD 1]
     IF FIRST :ANS = "BK  REPEAT LAST  :ANS
[BK 1]
     IF  FIRST :ANS = "RT  REPEAT LAST :ANS
[RT 1]
     IF FIRST :ANS = "LT  REPEAT LAST  :ANS
[LT 1]
     END
```

Modifications of these programs can serve other learning strategies. For example, in introducing young children to Logo, I frequently use the Tasmanian and Terrapin Robot turtles. But they require a "T" in front of each primitive. This sometimes slows a student's transferring learning between the robot and screen turtles. Simple modification of either of the above programs creates robot turtles that work with the screen turtle primitives. For example,

```
     TO ROBOT
     IF :ANS = [FD] TFT 50
     ...
```

HAPPY TIMES

Using these Logo modifications has made the children, their classroom teacher, and me happier. The children are more successful and pleased with their initial explorations in Logo. Being unable to type large numbers to wrap around, or get "Instant" responses to F and R keystrokes seems to encourage them to think more carefully about which commands to use.

The classroom teacher learns the Logo primitives and feels comfortable returning to the classroom and using Logo. I help the teacher develop strategies for continued use of Logo since my time is not occupied helping frustrated students.

These programs have been one small step in our lab towards creating an environment where all concerned are successful and productive in their efforts to learn and create together. This is far better than having a little boy crying because Logo is too hard for him.

C. M. Chmielowski is the Fairbanks (AK) District Computer Lab Co-ordinator, and teaches Logo classes to K - 10 students. In the summer, she teaches Logo classes for the University of Alaska at Fairbanks.

From the Editor

Fighting the Wheel

A friend once told me of her first experience with an automatic drive through car wash. She edged her car up to the entrance, rolled up her windows, and held on to the steering wheel with a death grip as the car was pulled into the path of the robot brushes, sprayers, and polishers.

She was terrified. She fought the towing mechanism as it moved her car from one operation to the other. The attendants kept tapping on the car window and shouting, "Let go of the wheel, lady!" But she refused to trust the giant washing machine. After emerging from the other end, she realized that her car was still in one piece, and so was she. Perhaps the machine worked OK.

A couple of weeks later, she had to get her car washed again. This time, she was a little more relaxed, and didn't fight the towing mechanism quite so much.

Lots of things affect us like that. Who can ever forget the anxiety of a first date, a first solo drive in a car, a first day at the first teaching job? What about when we first begin to teach with Logo?

How well I remember my first Logo attempts. One day, I would be too rigid and structured, "holding on to that steering wheel." The next day, I would be too loose. But, gradually, I found ways to teach with Logo which seemed to be effective. I began to relax and enjoy the experience with my students.

Steve Tipps often reminds anxious Logo teachers that Logo learning is developmental; it takes time. We all have to go "through the wash" a time or two before we feel relaxed and confident in what we are doing.

But, do not get the impression that, after taking a four-hour workshop or a three-hour course and teaching with Logo for a year, you can put your feet up. Logo learning has a special way of happening. It arrives quietly, in bits and pieces, as you try different things with your students. But it takes that effort on your part.

If you are having difficulty "letting go" as you begin your use of Logo, you have lots of company. Just don't be too hard on yourself. Time is on your side.

FD 100!

Tom

November 1984

NLXTRA

Logo Computer Systems Inc. (LCSI) announces the following list prices now in effect for their products. Apple Logo and Apple Logo II are $99 each. Apple Logo is available in Classroom Kits (10 language diskettes, a set of manuals, 10 reference cards, and a sample and tool kit diskette with documentation) for $399. Atari Logo is priced at $99 for the complete package, and $79 for the cartridge and reference card alone. SmartLogo for the Coleco Adam costs $74.95. For more information, contact LCSI, 220 Fifth Avenue, Suite 1604, New York, NY 10001, or 9960 Cote de Liesse Road, Lachine, Quebec, Canada H8T 1A1.

Terrapin Logo is now available in packs of multiple copies. A set of 10 language diskettes and reference cards, a utility diskette, and a set of documentation costs $299.95. A set with 20 language diskettes and reference cards costs $525.00. Terrapin also has announced a second utilities disk for Terrapin Logo for the Apple. The package contains 29 programs, including a driver for the Apple color plotter, a picture printing program for the Imagewriter, keyboard characters for the turtle screen, and a set of low resolution graphics tools. The cost is $19.95, and may be obtained directly from Terrapin, Inc., 222 Third Street, Cambridge, MA 02142. (617) 492-8816.

Jim Muller and the Young Peoples' Logo Association (YPLA) announce their 80 page Logo resource guide, The Logo Library, a listing of Logo versions, articles, books, groups, products, and projects. This publication contains comprehensive information on many aspects of Logo. To obtain a copy, send $7.95 plus $1.50 for shipping and handling costs to YPLA, PO Box 855067, Richardson, TX 75085.

It's never too early to think about what you will be doing next summer! Why not teach Logo at a summer camp? David Sachs of Computer Summers, Inc., announces vacancies for Logo teachers at some 15 locations across the country for the summer of 1985. Send your resume and references to David at PO Box 436, South Salem, NY 10590.

The New York State Association for Educational Data Systems holds its 19th annual conference November 18-20 at Kiamesha Lake, NY. Beginning and advanced Logo workshops are scheduled. For more information, contact Eileen Gress, Fairport Central Schools, District IMC, 140 Hurlburt Road, Fairport, NY 14450.

Vanderbilt University and IBM are sponsoring an innovative conference entitled "School of the Future" in Nashville, TN, on November 8-10. Five discussion groups meet simultaneously throughout the conference, each led by a panel of experts. Linton Deck, conference chairman, suggests that Logo teachers and supervisors might be interested in the groups on teaching and management. Panelists include Karen Billings, Karen Sheingold, and Roy Pea. For more information, write Future School Conference, Peabody College, Box 164, Vanderbilt University, Nashville, TN 37203. (615) 322-7311.

The fifth annual Florida Instructional Computing Conference will be held January 29-31, 1985 in Orlando. A preconference Logo workshop is scheduled, along with a number of Logo presentations during the conference itself. For more information, write Mr. David Brittain, Office of Educational Technology, Florida Department of Education, Knott Building, Tallahassee, FL. 32301. (904) 488-0980.

The second annual international conference, "Computers in Education" is scheduled for December 12-14, 1984 in Montreal, Canada. Featured speakers include Alan Kay, and Dan and Molly Watt. For additional information, write Computers in Education Conference, GEMS Conference and Consulting Services, PO Box 367, Snowdon, Montreal Canada H3X 3T6. (514) 735-1388.

The September 1984 issue of Classroom Computer Learning contains a stimulating Logo article by Pamela Sharp. "Using Logo to Shape Artistic Ideas" shows how to use Logo in art class. Four separate lessons are given for dual art and computer art work. CCL, Learning, Inc., 19 Davis Drive, Belmont, CA 94002.

BEST COPY AVAILABLE

George Cathcart, an education professor at the University of Alberta, sends the following procedures for printing inverse letters on the Apple Logo text screen.

```
     TO INVERSE :WHAT
     IF EMPTYP :WHAT [STOP]
     MAKE "WRD FIRST :WHAT
     WRITE :WRD
     TYPE CHAR 32
     INVERSE BUTFIRST :WHAT
     END

     TO WRITE :WRD
     IF EMPTYP :WRD [STOP]
     TYPE CHAR 128 + REMAINDER ASCII  FIRST
:WRD 64
     WRITE BUTFIRST :WRD
     END
```

Here are some examples to try.

```
INVERSE [ALL THIS IS HIGHLIGHTED]
TYPE [TRY \  ] INVERSE [THIS] TYPE [OUT!]
```

Eadie Adamson of the Nightingale - Bamford School in New York City sends this procedure to make erasing easier in Commodore Logo. She has used it successfully with her kindergartners.

```
     TO E :DISTANCE
     MAKE "COLOR ITEM 4 DRAWSTATE
     PENERASE
     BK :DISTANCE
     PC :COLOR
     END
```

Dear Editor,

With regard to the April 1984 NLXtra, I would like to add to your suggestion for editing more than one procedure in LCSI Logo. One need only type EDIT [SQUARE TRI TRI.ONE] to put all three procedures into the editor in one step. The MAKE statement (MAKE "ALL [SQUARE TRI TRI.ONE]) would only be useful if one intended to do extensive editing with those three procedures.

W.E.B. Corrigan
Beaconsfield, Quebec

>Thanks for your excellent suggestion.

Dr. Logo can be purchased for $64.85 plus $4.50 shipping from WCCI, PO Box 1028, Woodinville, WA 98072. If you call in your order, they issue you a credit of $2.50. (206) 481-2600.

Dear Editor,

I think that as teachers become more fluent in Logo and explore its expressive capacities, there may be a broader perception of its classroom applications. Teachers may become more comfortable using Logo as an exploratory medium and less inclined to present it ... as a structured, sequentially organized subject.

In this respect, I am personally in disagreement with the positive review of the Logo Guided Discovery Kits in the September 84 NLX. I think one of the glaring deficiencies of this approach is the lack of any single keystroke procedures to permit young children to explore the medium. In fact, I think this kit fosters precisely the wrong sort of implementation of Logo for teachers new to the language.

Robert J. Player
New York City public schools

>Regina Bowden responds: Thank you for your comments. The Logo Guided Discovery Kits were intended to be supplemental materials for teaching with Logo. Teachers new to the language need a variety of resources and specific suggestions to set up their learning environments. When combined with concrete activities and explorations, the Logo Guided Discovery Kits can aid creative learning.

School District No. 35 (Langley) has published a 79 page curriculum report which outlines planning for computer education within the district. Seeing the computer as a tool to free students to develop further their learning and thinking skills, the district computer committee based their plans accordingly. The section on the selection of computer languages is especially interesting, and makes a good case for Logo. Copies of this report are available for $10 US. Write School District 35 (Langley), Attn: Patrick McManus, 22259 48th Avenue, Langley, British Columbia V3A 3Z7, Canada.

The Educator 64 from Commodore is billed as "a computer designed for schools." It appears to be a Commodore 64 computer in the PET computer housing, with some additional features. It runs Commodore Logo and all other Commodore 64 software. Check with your local dealer for prices and availability.

BEST COPY AVAILABLE

Tipps for Teachers

by Steve Tipps

Number High, Number Low

Learning about equalities and inequalities is sometimes difficult for children. The TEST command with numbers provides another way to explore with Logo in mathematics class. By working with TEST and programming different consequences, children use the computer to test hypotheses about numbers. A number guessing game is one project which can be expanded in many ways.

QUEST FOR TRUTH

The computer obeys consequences, but setting up the TEST and the results is a task for the problem solver. Begin in the immediate mode with a simple TEST. (For this column, I am using Terrapin Logo. In most cases, the conversion to LCSI Logo is straightforward.)

```
TEST 9 = 9
```

Many IFTRUE consequences are possible

```
IFTRUE PRINT [ 9 = 9 ]
9 = 9
IFTRUE PRINT [ TRUE ]
TRUE
IFTRUE REPEAT 9 [ PC 3 RT 180/9 PU  FD
50 PD FD 50 PU BK 90 ]
```

as are many IFFALSE consequences.

```
IFFALSE PRINT [ FALSE ]
IFFALSE [CRAZY.STAR]
```

In the case of TEST 9 = 9, the result of the TEST is true and only the IFTRUE consequences happen. Nothing at all happens when the IFFALSE consequences are typed. The TEST sets a signal of either TRUE or FALSE. That signal, or flag, stays set up until a new TEST is made.

Students can try writing TESTs and consequences in the immediate mode for any numbers and talk about the expected results.

```
TEST 100 > 30
                        "That's true."
IFFALSE PRINT CHAR 7
                        "Nothing happens."
IFTRUE PRINT CHAR 7
                        "It beeps!"
IFFALSE PRINT [FALSE]
                        "Nothing happens."
IFTRUE PRINT [ YES, IT IS TRUE ]
                        "This should print."
PRINT [ 100 > 30 ]
                        "This always prints."
```

If the TEST is changed to TEST 100 < 30, the IFFALSE consequences are executed. PRINT [100 > 30] is unconditional and always executes. Rather than writing in the immediate mode, it is much handier to write a procedure containing consequences.

```
TO IS.IT.TRUE?
IFTRUE PRINT CHAR 7
IFTRUE PRINT [YES, IT IS TRUE.]
IFTRUE BG 3
IFTRUE PC 4 SQUARE
...
END

TEST 40 = 40
IS.IT.TRUE?
YES, IT IS TRUE.
...
```

IS.IT.TRUE? and a companion procedure IS.IT.FALSE? can be expanded with more and more consequences.

NUMBERS GREAT AND SMALL

After testing numbers many times and in many conditions, students should see that three stiuations are possible when comparing two numbers. They can be equal or unequal. If they are unequal, one or the other will have the larger value.

COMPARE might be a good name for a procedure which will compare any two numbers. Prepare for COMPARE by talking about all the possibilities. A description of COMPARE might be:

Find out if the first number is equal to the second.

If they are equal, print an equality statement and stop.

If they are not equal, find out if the first is larger.

If the first is larger, print that it is.

If the first is not larger, print that the second is larger.

Tipps continued

An English description in logical steps is sometimes called pseudocode. Some people use flowcharts to do the same kind of planning. The actual procedure does exactly what is described in the pseudocode.

```
     TO COMPARE :FIRST :SECOND
     NODRAW
     TEST :FIRST = :SECOND
     IFTRUE  PRINT1  :FIRST  PRINT1  ['  IS
EQUAL TO  '] PRINT :SECOND
     IFTRUE STOP
     TEST :FIRST > :SECOND
     IFTRUE  PRINT1  :FIRST  PRINT1  ['  IS
GREATER THAN '] PRINT :SECOND
     IFFALSE  PRINT1  :SECOND PRINT1 ['  IS
GREATER THAN '] PRINT :FIRST
     END
```

The important statements of relationships between the numbers are in the procedure, but no one gives this procedure any prizes for excitement. A number guessing game might be a more stimulating way to play with the inequalities.

GETTING CLOSER

A simple guessing game would set the number and let you try to guess it.

```
TO TRY :NUMBER
TEST :NUMBER = 14
IFTRUE PRINT [THAT IS IT.]
IFFALSE PRINT [THAT IS NOT IT.]
END
```

Even if the range were restricted to numbers from 1 to 50, the process of guessing would be fairly laborious.

```
TRY 30
THAT IS NOT IT.
TRY 15
THAT IS NOT IT.
TRY 9
THAT IS NOT IT.
```

Forty-six tries later you might guess the correct number. In exasperation, you might even be tempted to peek at the number by looking in TRY.

It would be helpful if students find out immediately that 9 is too low and 30 is too high. Also, 15 is closer than 30. This game gives you no information about the different numbers. Using ideas from COMPARE makes the game much better by providing informative feedback about which number is too high or too low.

```
TO GUESS :NUMBER
TEST :NUMBER = 77
IFTRUE PRINT [THAT IS RIGHT.]
IFTRUE PRINT [77 IS THE NUMBER.]
IFTRUE STOP
TEST :NUMBER < 77
IFTRUE PRINT [ THAT IS TOO LOW. ]
IFFALSE PRINT [THAT IS TOO HIGH.]
END
```

Now the guesser gets information about the guess.

```
GUESS 89
THAT IS TOO HIGH.
GUESS 31
THAT IS TOO LOW.
GUESS 77
THAT IS RIGHT.
77 IS THE NUMBER.
```

Information about the guess makes the game more interesting, but not exciting. Programming other consequences such as coloring the background or drawing pictures would make the game more entertaining.

A guessing game might proceed through six or eight stages as students refine and add to it. The teacher serves as a programming consultant in the project.

GUESS is only a skeleton program for students and teachers to improve into an educational game on numbers. Starter programs like this should stimulate problems because they don't do everything you want them to do.

FIXING IT

One thing which this program doesn't do is give you an easy way to change the mystery number. It is possible to change the number by editing GUESS and putting in a new mystery number. But you must insert the new number in three separate places. You want an even easier method. Another way to change the number is to assign a value to IT with a MAKE statement.

```
MAKE "IT 35
```

Now, GUESS can be changed to GUESS.IT, to use the value of assigned to the variable IT. Of course, the variable IT must appear in three separate places in the procedure.

```
     TO GUESS.IT :NUMBER
     TEST :NUMBER = :IT
     IFTRUE PRINT [THAT IS RIGHT.]
     IFTRUE  PRINT1  :IT PRINT  ['  IS  THE
NUMBER.']
```

Tipps continued

```
     IFTRUE STOP
     TEST :NUMBER < :IT
     IFTRUE PRINT [THAT IS TOO LOW.]
     IFFALSE PRINT [THAT IS TOO HIGH.]
     END
```

What if you want to play GUESS.IT solitaire? The excitement pales if you have to MAKE it and GUESS.IT, too. Logo can randomly choose its own mystery nuumber.

```
     TO SET.IT
     RANDOMIZE
     MAKE "IT RANDOM 100
     PRINT  [TRY TO GUESS THE  MYSTERY
NUMBER.]
     PRINT  [IT IS A NUMBER FROM 0  THROUGH
99.]
     END

     SET.IT
     TRY TO GUESS THE MYSTERY NUMBER.
     IT IS A NUMBER FROM 0 THROUGH 99.
```

SET.IT sets a new number for IT, and the guessing game can start again. RANDOMIZE is used to reset the random number list.

NUMBERS UP

COMPARE, TEST, and GUESS.IT might also be used for an estimation game. Announce a target number, and several arithmetic problems. A simple procedure with PRINT might display this information.

```
     THE TARGET IS 357

     WHICH PROBLEM GIVES THE
     ANSWER CLOSEST TO 357?
     42 x 8
     21 x 18
     52 x 7
     11 x 33

     WHICH ANSWERS ARE TOO HIGH?
     WHICH ANSWERS ARE TOO LOW?
     ESTIMATE, THEN USE LOGO TO CHECK.
```

Logo provides many ways to work on this problem - from using GUESS.IT to calculating in the immediate mode.

```
     MAKE "IT 357
     GUESS.IT 21 * 18
     THAT IS TOO HIGH.
     COMPARE 357 52*7
     364 IS GREATER THAN 357
     PRINT 11*33
     363
     PRINT 11*33 - :IT
     6
```

Estimation skills can reveal much about students' understanding of arithmetic. Logo is a tool for testing estimates and increasing understanding of numbers. Whether the number is too high, too low, or just right, Logo helps everyone win by learning about numbers.

Next month, Numbers in Between.

Steve Tipps is the West Professor of Education at Midwestern State University in Wichita Falls, TX. He presents Logo workshops for school systems throughout the United States.

NLXionary

A Lectionary of Selected Logo Readings

by Griff Wigley

This month's selections are from panelist papers written for Logo 84, the National Logo Conference held at MIT this past June.

"Creating Logo Cultures," by Dan Watt

>Watt's contention is that Logo needs a supportive "culture" to be successful in the way that reading and writing have been. By culture, he means an environment in which people regularly engage in activities related to the skill in a multitude of ways. He thinks that expecting teachers to create successful Logo experiences for students after only a few days of learning Logo is analogous to expecting teachers to teach reading and writing successfully after only a few days of learning to read and write.

>However, he cites several examples in which Logo utilization has gone beyond the mechanics of learning the language, showing that "Logo can indeed serve as a catalyst and an organizing principle for the kind of human culture we are all trying to create ... rich with experiences, ideas and people that break down the barriers between disciplines, between teacher and learner, and between people with different learning styles."

NLXionary continued

>Watt doesn't mention that this kind of environment can be created without one smidgeon of Logo, which I find slightly disturbing. His reading / writing analogy and closing statements seem to elevate Logo's importance needlessly. I prefer to differentiate Logo from the philosophy behind it, seeing it instead as one of many possible threads in the current fabric of effective learning environments.

"About That Rose Garden: Remarks on Logo, Learning, Children and Schools," by William Higginson

>Philosophy delivered with delightful wit by the editor of Logophile. Higginson lampoons research that attempts to measure Logo's impact in behavioristic and quantitative terms. He reminds us that Logo is a current battleground "for an old philosophical dispute," and cites writers in education and in other disciplines whose philosophy is parallel to that of Logo. Rather than simply asking, "What do children learn from Logo?" he prefers to ask, "Is it the case that Logo can be used to aid the intellectual and social growth of children in the manner described by Papert?" The man has data to answer the question.

"So What Do Children Learn with Logo?" by Rina Cohen

>A dialogue between a Logo educator (the author) and a Logo mathematician illustrating issues involved in evaluation, some typical negative outcomes involving the implementation of Logo, and some possible solutions. One interesting idea for evaluating Logo-related learner outcomes, even with a group of students who are learning Logo from a teacher only recently afflicted with "Logo fever" is to measure specific cognitive skills such as map reading and directionality. Trying to measure Logo's potential for fostering a love of learning, a sense of mastery and power, cooperation, and all the other components that contribute to a student developing his or her full ability is too risky because it depends too much on the individual teacher's personal characteristics and philosophy of education.

Griff Wigley is a co-owner of Family Computing, Inc., a facilitator in the Faribault (MN) Public Schools, and a school board member of Prairie Creek Community School.

Logo 84 Proceedings Available

NLX readers may obtain a copy of the Logo 84 conference proceedings which contain the papers mentioned in this month's NLXionary, in addition to many other papers and abstracts. Also included is an extensive Logo bibliography containing listings of books, periodicals, and articles. To obtain your copy, send a check for $15 US to MIT Logo Group/Publications, NE43-438, 545 Technology Square, Cambridge, MA 02139.

TurtleTips

by
Donna Lanyi
and Jane Toth

What takes the most time in writing a procedure? TYPING IT IN!! By now, you have noticed that many students spend a great deal of time "hunting and pecking." Keyboarding skills do play a major role in developing computer skills.

TEACHER HINTS

A useful teaching aid that we have developed is a posterboard size picture of the keyboard. We have color coded the home row keys to match a set of color coded hands that accompany the picture. We also supply students with a desktop laminated keyboard. This allows students to practice keyboarding skills without using computer time.

We use keyboarding practice books for children that start with the home row keys and build. Sometimes we use old typing books for exercises. Our students also practice typing procedures so that they familiarize themselves with the most commonly used Logo keys.

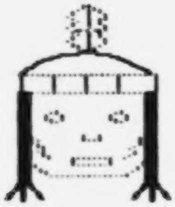
If you can find an old typewriter, place it in a corner of your room and allow students to use it for practice. Perhaps you could develop a learning center for these activities.

Commercially produced practice keyboards are available for most popular computers from Computer Practice Keyboard Company, 616 9th Street, Union City, NJ 07087, for $4.95 each. The price drops to $3.00 if you order 12 or more. Be sure to specify your computer make and model.

TurtleTips continued

PROCEDURE OF THE MONTH

For November, have your students design a picture to depict Thanksgiving. Ask your students to match our Indian face, or create one of their own. If you are interested in a free listing of this procedure, please send us a self addressed stamped envelope.



We invite elementary teachers and students to send procedures or ideas to be shared. Donna Lanyi and Jane Toth, Beall Avenue School, 716 Beall Avenue, Wooster OH 44691.

Donna Lanyi and Jane Toth use Logo in their elementary classrooms in the Wooster (OH) City Schools.

Logo Teachers Note

Are you looking for a new dimension to your teaching this year? The Logo Class Penpal Network may be just what you are looking for. The Logo Class Penpal Network matches participating teachers with counterparts in various regions of the USA, Canada, or one of several overseas countries according to grade level, computer compatibility, Logo experience and interests. During the school year, students and teachers exchange Logo ideas, procedures, and projects. Of course, many subject areas can be worked in as the exchange continues: handwriting, geography, social studies, and mathematics, just to name a few.

If you are interested in participating in our network, send us a self addressed stamped envelope and we will send you our free application kit. We anticipate an outstanding year; we hope you will be part of that year. Logo Penpals, School of Education, University of Virginia, 405 Emmett Street, Charlottesville, VA 22903-2495.

(Ed. note: The NLX Logo Class Penpal Network has a new home! It has been adopted by the University of Virginia's School of Education.)

Listful Thinking

by

Glen Bull and Paula Cochran

Count Your Blessings

For thousands of years, people in cultures around the world have celebrated the time of harvest. In America, Governor Bradford was the first to declare a day of Thanksgiving among the colonists of Plymouth in November, 1623. Everywhere, such a celebration causes people to reflect on their good fortune, families, favorite foods, (football, perhaps), -- and the list goes on. Which brings us to Logo (subtle, huh?).

This month, we would like to help you explore how to COUNT items in a list and ADD items to a list. We'll show you how to INSPECT your lists, and use them to make Thanksgiving story starters.

WHAT'S IN A LIST?

As you recall from last month, Logo can be used to MAKE a list as follows:

```
MAKE "FEAST [TURKEY CORN YAMS]
MAKE "PILGRIM [PRISCILLA MILES]
```

You can ask Logo to print out the names of your lists and their contents with the command PONS (Print Out NameS). (For MIT Logo, use PO NAMES.)

```
PONS
"FEAST IS [TURKEY CORN YAMS]
"PILGRIM IS [PRISCILLA MILES]
```

As we saw last month, we can also PRINT the contents of a list.

```
PRINT :PILGRIM
PRISCILLA MILES
```

AT LAST COUNT

Let's suppose we want to know how many items are in a list without having to count them ourselves. Logo can do this and then tell us the number, using the COUNT primitive. (If you are using Version 1 of Terrapin or Krell Logo, COUNT must be defined as a procedure such as the one at the end of this column.)

Listful Thinking continued

COUNT outputs a number. We must tell Logo what to do with that number. For example, we could PRINT it:

```
PRINT COUNT :FEAST
3
PRINT COUNT :PILGRIM
2
PRINT (COUNT :FEAST)+(COUNT :PILGRIM)
5
```

MORE PILGRIMS AND MORE TOOLS

In previous columns, we have been encouraging you to use the procedure PICK as a tool. ADD and INSPECT are two more procedures which can be used in this way. Your students can use them much as they would use REPEAT or other primitives. ADD can add items to an existing list with the LPUT command.

```
TO ADD :ITEM :LIST
MAKE :LIST LPUT :ITEM THING :LIST
END
```

This procedure can add the name SUE to the list of PILGRIMs.

```
ADD "SUE "PILGRIM
```

To check,

```
PRINT :PILGRIM
PRISCILLA MILES SUE
```

INSPECT prints the contents of a list in a neat, one-per-line format. (See a slightly different procedure at the end of the column for Terrapin and Krell.)

```
TO INSPECT :LIST
IF :LIST = [] [STOP]
PRINT FIRST :LIST
INSPECT BUTFIRST :LIST
END

INSPECT :PILGRIM
PRISCILLA
MILES
SUE
```

Here's a procedure that makes it easy to add to the list of pilgrims and count them, too. The command READLIST waits until you type a name on the keyboard. Then ADD accepts the name and adds it to the list of pilgrims. (If you are using Terrapin or Krell, substitute REQUEST for of READLIST in the procedure.)

```
TO ADD.PILGRIM
PRINT [NAME A NEW PILGRIM]
ADD READLIST "PILGRIM
PRINT []
COUNT.PILGRIMS
END

TO COUNT.PILGRIMS
PRINT [HERE ARE THE PILGRIMS:]
INSPECT :PILGRIM
PRINT []
PRINT  (SENTENCE  [THERE  ARE]  COUNT
:PILGRIM [PILGRIMS NOW!])
END
```

Here's what happens when ADD.PILGRIM is used.

```
ADD.PILGRIM
NAME A NEW PILGRIM
TOM

HERE ARE THE PILGRIMS:
PRISCILLA
MILES
SUE
TOM

THERE ARE 4 PILGRIMS NOW!
```

You could make a similar procedure to add to the list of foods.

```
TO ADD.TO.FEAST
. . .
. . .
END
```

You may have noticed that one or two of the lines in the ADD.TO.FEAST procedure have been omitted. In a Logo-like fashion, we invite you to consult ADD.PILGRIM and develop your own personal ADD.TO.FEAST procedure. If you are emphasizing nutrition, your class could establish a separate list for each food group, for example.

ALL THE FIXINGS

It's easy to make a mistake typing a name. Suppose that, instead of TOM, you actually wanted TIM in the list of pilgrim names. In Logo, you can edit lists of names, just as you can edit procedures. Just type EDNS. (With Terrapin and Krell, type EDIT NAMES.)

When you edit names, you go to the editor. Instead of a procedure, you see something that looks like this:

```
MAKE "FEAST [TURKEY CORN YAMS]
MAKE "PILGRIM [PRISCILLA MILES SUE [TOM]]
```

Listful Thinking continued

Now use the edit keys to move over to TOM and fix the spelling, just as though you were editing a procedure. Then, exit the editor in the usual way to define the edited lists.

(By the way, [TOM] appears in the pilgrim list instead of just TOM. This is because ADD.PILGRIM is using READLIST to provide the input to ADD. This allows those with names such as MARY JANE to be ADDed to the pilgrim list.)

THE FIRST THANKSGIVING

After the children in your class add their names to the list of pilgrims, they can be in a story. As we saw last month, the PICK procedure is used to pick an item randomly from a list. (See listing at the end of this column.)

```
     PRINT PICK :PILGRIM
     MILES
```

In a sentence, PICK works like this:

```
     PRINT (SENTENCE PICK :PILGRIM [WAS  IN
CHARGE OF THE FEAST.])
     TIM WAS IN CHARGE OF THE FEAST.
```

What can you do with a list of pilgrims and a feast menu? After your children have developed the lists, they can use a short procedure (such as the one which follows) to generate individual story starters. The story starters can be used as a basis for oral story-telling, a writing assignment, an art project, etc.

```
     TO STORY
     PRINT (SE [THERE WERE] COUNT  :PILGRIM
[PILGRIMS IN THE COLONY.])
     PRINT (SE PICK :PILGRIM [WAS IN CHARGE
OF THE FEAST.])
     PRINT (SE [EVERYONE BUT] PICK :PILGRIM
[LIKED] PICK :FEAST [.])
     END
```

A story starter might look like this.

```
     STORY
     THERE WERE 3 PILGRIMS IN THE COLONY.
     SHEILA WAS IN CHARGE OF THE FEAST.
     EVERYONE BUT DICK LIKED TURKEY .
```

Each story starter has different names. As more pilgrims are added to the list, the number in the story changes, too.

```
     ADD.PILGRIM
     NAME A NEW PILGRIM
     KEELAN

     HERE ARE THE PILGRIMS:
     MILES
     DICK
     SHEILA
     KEELAN

     THERE ARE 4 PILGRIMS NOW!
```

With the new pilgrim added, the story starter might change to something like this:

```
     STORY
     THERE WERE 4 PILGRIMS IN THE COLONY.
     MILES WAS IN CHARGE OF THE FEAST.
     EVERYONE BUT KEELAN LIKED YAMS .
```

We hope that you and your students enjoy COUNTing and ADDing your blessings this month.

Next month: Making a List and Checking It Twice!

PROCEDURES

Here is last month's PICK procedure:

```
TO PICK :LIST
OUTPUT ITEM 1 + (RANDOM COUNT :LIST) :LIST
END
```

If you are using Version 1 of Terrapin or Krell Logo, the following procedures are needed with PICK.

```
     TO ITEM :NUMBER :LIST
     OUTPUT ITEM.LOOP :NUMBER :LIST 1
     END

     TO ITEM.LOOP :NUMBER :LIST :TALLY
     IF  :LIST = [] OUTPUT (SE  [THERE  ARE
ONLY] :TALLY [ITEMS IN THIS LIST.])
     IF :NUMBER = :TALLY OUTPUT FIRST :LIST
     OUTPUT   ITEM.LOOP   :NUMBER   BUTFIRST
:LIST :TALLY + 1
     END

     TO COUNT :LIST
     OUTPUT COUNT.LOOP :LIST 0
     END

     TO COUNT.LOOP :LIST :NUMBER
     IF :LIST = [] OUTPUT :NUMBER
     OUTPUT   COUNT.LOOP   BUTFIRST   :LIST
:NUMBER + 1
     END
```

Listful Thinking continued

This is INSPECT for Terrapin and Krell Logo.

```
TO INSPECT :LIST
IF :LIST = [ ] STOP
PRINT FIRST :LIST
INSPECT BUTFIRST :LIST
END
```

Glen Bull is a professor in the University of Virginia's School of Education, and teaches Logo courses at both the graduate and undergraduate level. Paula Cochran is a reformed English major who studied linguistics at the University of Cambridge. She is now a speech-language pathologist working with language-disabled children.

Teacher to Teacher

Logo Exploration: An Adventure for Middle School Students

reviewed by Jim Fry

"You are about to begin an adventure. Imagine that you are sailing to a chain of islands in the Pacific Ocean. The purpose of the trip is to explore the islands and learn all you can about what is there: mountains, rivers, volcanoes, birds, other animals, flowers and trees."

Thus begins <u>Explorer's Guide to Apple Logo</u>. The authors state that their book is a guide to Logo, like a map which shows where places are located and gives some information about them. But, as with a map, you still need to visit the places yourself and make your own discoveries. And, as with a guide book, you might not want to proceed in the order of the pages; you might want to skip around.

WHAT A BEGINNING

What an exciting way for a teacher to introduce Logo! After I read the first chapter, I was brainstorming with myself about all the ways I could tie in the explorer idea with activities for my students to use both in the classroom and at home.

The book itself is a six chapter workbook, with an appendix of eight sections dealing with such things as a summary of Logo commands, glossary, information on handling disks, using a printer, and other information for teachers.

The six chapters are:

1. Introduction - discusses the idea of exploration and sets the stage
2. Activities without the Computer - an important chapter for precomputer work, especially for classes with limited computer time
3. Commanding the Turtle - suggests beginning activities for turtle graphics
4. Procedures - includes working in the editor and saving procedures
5. Shapes - explores geometric concepts and introduces the idea behind the Total Turtle Trip Theorem
6. Making Designs - expands on Chapter 5 and introduces superprocedures

The book is organized in a helpful manner. Reference pages have colored borders to make them easy to spot. Activity pages with ideas and suggestions for things to try make up the bulk of each chapter. The more important activities are marked clearly. Challenger activities for more advanced students are also included. There is plenty of room for writing and sketching on the pages.

SPRINGBOARDS AND STARTING POINTS

Many ideas in the book can become springboards for additional activities. For example, after the students are introduced to the turtle, the concept of degrees for use with the RIGHT and LEFT commands is developed. The authors indicate how map making and navigation employ degrees, how the Air Force uses the numebrs on a clock to judge position, and how degrees relate to these numbers. Directions to make a turtle clock are included. There are also additional examples in the back of the book for teachers to use.

This part of the book would be a perfect addition to a unit on compass use. Mapping activities could be developed by teacher and students using the turtle to draw treasure maps or mystery island maps with a compass rose to show direction. Students could then give compass directions and turtle steps (old pirate maps used paces, I'm told) to friends to find the hidden treasure or the mystery location.

Teacher to Teacher continued

FOR WHOM?

Upper elementary or middle school students would be able to use Explorer's Guide to Apple Logo on their own, with guidance from the teacher. The reading level is too difficult for most students below fifth grade. For teachers in lower grades, the book could serve as a source of ideas for simpler student activities.

Explorer's Guide to Apple Logo lives up to its name as a guide book to help students and teachers become explorers in the exciting world of Logo. A teacher's manual and a review set of 15 spirit duplicating masters are also available.

Explorer's Guide to Apple Logo (LCSI Version) by Joan Moscovitch, Peter von Mertens, and Maggie Neville Holmes, $12.95, Hayden Book Company, 10 Mulholland Drive, Hasbrouck Heights, NJ 07604. (201) 393-6000. An edition for MIT Logo is available also.

If you have comments, questions, or concerns about this review, please contact Logo Information For Teachers (LIFT) at PO Box 5396, Plymouth, MI 48170.

Jim Fry teaches with Logo at Novi Elementary School in Novi, MI, and is a cofounder of LIFT.

NLX Notes

It's not too early to start thinking about Christmas gifts for special people. How about a gift subscription to the NLX for a teacher relative or friend? Send $25 and a written message to NLX PO Box 5341, Charlottesville, VA 22905. We will mail a special Christmas gift notification which includes your written message, and start the subscription with the January issue.

The next edition of the NLX Logo Research Directory will be published soon! If you are conducting educational research relating to Logo, you should be listed. Please send your name, address, and a summary of your research interests and activities to NLX Research Directory, Attn: Regina Sapona, PO Box 5341, Charlottesville, VA 22905.

Q and A

by
Jim McCauley

Q. I'm really puzzled by OUTPUT and its use in recursive procedures. I was working with a procedure like:

```
TO ANY.GREATERP :N :LST
IF EMPTYP :LST [OUTPUT "FALSE]
IF (FIRST :LST) > :N [OUTPUT "TRUE]
OUTPUT ANY.GREATERP :N BUTFIRST :LST
END
```

which checks to see if there are any numbers greater than N in a list called LST. What took me a long time to figure out is this: Not only must you arrange for the procedure to OUTPUT TRUE and FALSE, but you must also OUTPUT the recursive call to ANY.GREATERP! Why? This seems to me to be one of the features of Logo that is not really well documented.

<[-?-]>

A. Let me open my response by agreeing immediately with your last comment. While there are many examples of this feature of LCSI and MIT Logos in the documentation supplied, it is left to the user to "read between the lines" to pick up on this technique. Indeed, most of the trade books on Logo give it only the lightest treatment. The authors assume that it is "logical" (and it is ...), but I agree with you that it is not intuitively obvious.

Let me try to make it a bit more comprehensible. Take a procedure like:

```
     TO ADD.N :N :LST
     IF EMPTYP :LST [OUTPUT []]
     OUTPUT SENTENCE (FIRST :LST) + :N
ADD.N BUTFIRST :LST (all on one line)
     END
```

This adds the number N to every number in the list LST and outputs the modified LST when the job is done. Its operation is classically recursive: until LST is empty, it "holds on" to the first number in LST, adding N to it, then creates another "copy" of itself to send the BUTFIRST of LST for further processing.

Each of the copies creates other copies until LST is empty. The last copy receives an empty LST and outputs it.

Q and A continued

Like STOP, OUTPUT halts the execution of a procedure. But, unlike STOP, it returns some information to the procedure that called it into action.

THE CHAIN GANG

Imagine all of these copies of ADD.N as a chain, each copy linked to the one that called it and the one it called. When LST is empty, the chain "bottoms out," and an enpty LST is output to the next-to-last copy of ADD.N.

Now that copy OUTPUTs a sentence made of two things: the number to which it added N and the enpty LST it received from the last copy of ADD.N.

The LST it outputs is treated by the copy "above" it in a similar way, a process which continues until the "top" copy outputs the reconstructed LST to the primitive or procedure that called ADD.N in the first place. Thus,

```
SHOW ADD.N 4 [21 14 52]
```

produces

```
[25 18 56]
```

(If you use MIT Logo, change the first line in ADD.N to IF EMPTY? :LST OUTPUT [] or IF :LST = [] OUTPUT [], and test it with ADD.N 4 [21 14 52]. There is no SHOW primitive in MIT Logo.)

I hope this helps you to understand why the last line of ADD.N contains OUTPUT. It is necessary to return the processed LST to all the "higher" copies of ADD.N so that the entire LST can be reconstructed. If you'd like more practice in understanding how this works, Harold Abelson has a nice illustration in Section 8.1 of his Logo primer, Apple Logo (for LCSI), or Logo for the Apple II (for MIT Logo), published by Byte/McGraw Hill.

BACK TO ANY.GREATERP

In your ANY.GREATERP example, things are only slightly different. ANY.GREATERP is not trying to preserve the structure of a list that it will later reassemble; instead, it is simply taking the list apart and looking for a number greater than N. If it finds one, it tells the world by outputting TRUE and dropping the task. If it gets to the end of LST without finding what it's looking for, it outputs FALSE and halts. So it seems a bit unnecessary to OUTPUT the recursive call to ANY.GREATERP, since we don't expect to reconstruct the list that has been torn apart.

ANY.GREATERP is a "tail recursive" procedure, a special type that resembles a loop. It involves a system programmer's trick. Usually, recursive procedures eat up memory like crazy, since there are all those copies lying about, waiting for their "offspring" to return to them with some output.

LOGO INTERPRETED

Often, Logo designers save memory by making the Logo interpreter smart enough to detect when tail recursion is called for. These smart interpreters simply handle such procedures as though they were loops. In order to get the trick to work, though, the syntax must remain consistent, so OUTPUT must appear in the line that makes the recursive call.

Incidentally, some Logo interpreters are smarter than others with respect to this. I've tested all the Logo implementations from LCSI, and they treat both tail recursive operations (procedures that output) and tail recursive commands (procedures that don't output) as I have described. The MIT versions for the Apple from Krell and Terrapin treat tail recursive commands in this way, but not tail recursive operations.

Usually, this is not a serious limitation, since you'd have to have more than eighty items in a list that was being processed recursively before you ran past the interpreter's capacity to store the previously created copies.

I hope this has helped you understand better all that happens when you use OUTPUT recursively. Good luck!

(Ed. note: If you have questions about Logo programming techniques or the thinking which leads to Logo programming, send them to NLX Q & A, PO Box 5341, Charlottesville, VA 22905. Each month, Jim McCauley answers selected questions in this column.)

Jim McCauley is a Coordinator of Computer Education for the Santa Clara (CA) County Office of Education, and has written Logo articles for many national publications.

前 100

by
Hillel
Weintraub

Some very exciting developments in the growth of Computer Assisted Learning (CAL) have occurred in Japan in the past few months.

I attended a fantastic CAL meeting in Osaka, organized by Shunichi Yamazaki. Hands-on sessions with the new Fujitsu Logo were presented to an audience of mostly beginners by Masaru Takahashi.

Takito Totsuka and I made presentations also. Takito is a primary school teacher who is using Logo to increase his students' understanding of natural phenomena such as leaf structure and movement of planetary satellites.

AN ENTHUSIASTIC REACTION

These presentations got the audience members very enthusiastic about the real potential of Logo in CAL. Much discussion centered around Logo itself. Is it Logo that is magical? Or, perhaps the magic comes from how it is used by creative teachers to help learning take place.

We issued ourselves this warning:

"We must remember that it is not Logo itself that we are advocating, but what we feel Logo represents: a learning environment in which students can discover important things about themselves and the world."

FUJITSU LOGO

Here is more information about the new Fujitsu Logo. It runs on the Fujitsu 7 series of computers. The newest one, the FM-77, is quite a machine: two drives, a 14" RGB monitor, and a detached keyboard for about $1000 discounted. A major point of the Fujitsu advertising campaign is the bundling of Logo: that is, it comes with the machine. This is pretty amazing, considering that most people in Japan had never heard of Logo a few months ago!

Fujitsu Logo was created by Logo Computer Systems Inc. (LCSI) and is similar in operation to Apple Logo II. (Ed. note: for a review of Apple Logo II, see NLX Sept 1984.)

The Fujitsu keyboard has been especially designed to make input of common Logo symbols such as quotes and dots very convenient. I believe we will be seeing more of this type of user consideration in the future as Logo is designed for other specific systems.

Some other special features of Fujitsu Logo include: a 3 channel music system; dual 68B09/Z-80 microprocessors for speedy turtle graphics; over 4000 nodes available; an editor for 32x16 sized shapes; 4 turtles; and the unique ability to handle Japanese characters (Kanji) by treating them as turtle shapes!

Of course, Logo is not meant to use Kanji in word processing, but to be able to use them in graphic applications or in translation games is a nice addition.

Fujitsu Logo has an enhanced capability to handle property lists, including several special property list primitives. The use of Logo for keeping records is emphasized.

There is no news about the release of Fujitsu Logo in the USA, but I'm sure that the features on this extraordinary version will affect the design of future Logo systems released everywhere!

If you have any questions about Logo in Japan, write to me at Doshisha International High School, Tatara, Tanabe-cho, Tsuzuki-gun, Kyoto-fu, Japan 610-03.

Next month, I'll bring you up to date on Epson's mighty Logo for the QX-10. Until then, MAE 100!

Hillel Weintraub is a teacher at Doshisha International High School in Kyoto-fu, Japan, and is the president of the Society for Microcomputing in Life and Education (SMILE).

64K for the Apple III

Have you been wanting to run Logo on your Apple III, but could not get it to work, even in the Apple II emulator mode? This because only 48 K can be addressed. There is a new emulator board now available from Titan Technologies which has 64 K of addressable memory for the Apple III. For the first time, this makes it possible to run the "regular" Logos on the III. There is also a configuration available with 128 K. For more infomation, contact Debbi Kennard, Titan Technologies, Inc., 310 West Ann Street, Ann Arbor, MI 48104, or call (313) 662-8542.

Logo Disserts:

Dissertations Dealing with Logo

by Barbara Elias

Logo learning and geometry concepts naturally go together. Turtle geometry is often used to introduce Logo. Many geometry concepts can be explored in this interactive animated environment. When considering geometry concepts for the young child, however, we must be mindful of how the child tends to view shapes, lines, and angles.

For example, we see a square as a square - rigid and fixed. Young children view lines and shapes from a more topological perspective; that is, they see the shapes as being flexible and able to bend rather than as rigid and fixed.

THE QUESTIONS

Is it possible that a Logo learning experience can influence the way young children think about geometry and perhaps lead to more systematic thought processes? L. P. Rieber's masters thesis examines both this question and a broader one: "Can Logo learning influence the developmental characteristics of young children?" The latter is the kind of question that Papert challenges us to examine.

Now that the microcomputer and Logo are classroom tools, questions like this can be investigated in the regular classroom setting. It was in such a setting that second graders explored the concepts of angle, degree, measurement, and the Total Turtle Trip Theorem. The teacher, aided by fifth grade tutors, used a guided discovery approach during the three month long study. Children were instructed in the basic turtle commands, geometry, and measurement. There were encouraged to "play turtle" as they worked.

THE RESULTS

Pretest and posttest data were collected from Piagetian measures, a teacher designed geometry test, and interviews. There was a significant difference in the performance on children in the experimental group on all measures. Logo was a new experience for the children in this study. It was also an enjoyable experience.

The results suggest that Logo learning can have an impact on the development of systematic thought in the young as well as provide a means of examining developmental characteristics from an additional perspective. More comprehensive studies in this area are needed.

>>>Rieber, L. P. (1983). The Effect of Logo on Increasing Systematic and Procedural Thinking According to Piaget's Theory of Intellectual Development and on its Ability to Teach Geometric Concepts to Young Children. (Unpublished Masters Thesis, University of New Mexico.)

(Ed. note: Each month, Barbara Elias highlights a dissertation or thesis dealing with Logo. She is accepting copies of recent research results for the NLX Dissertation and Thesis Repository. Reports may be mailed to her in care of NLX, PO Box 5341, Charlottesville, VA 22905.)

Barbara Elias is an assistant professor in the Education Department of Virginia State University in Petersburg, VA, and a doctoral candidate at the University of Virginia.

Michigan Logo Users Group Formed

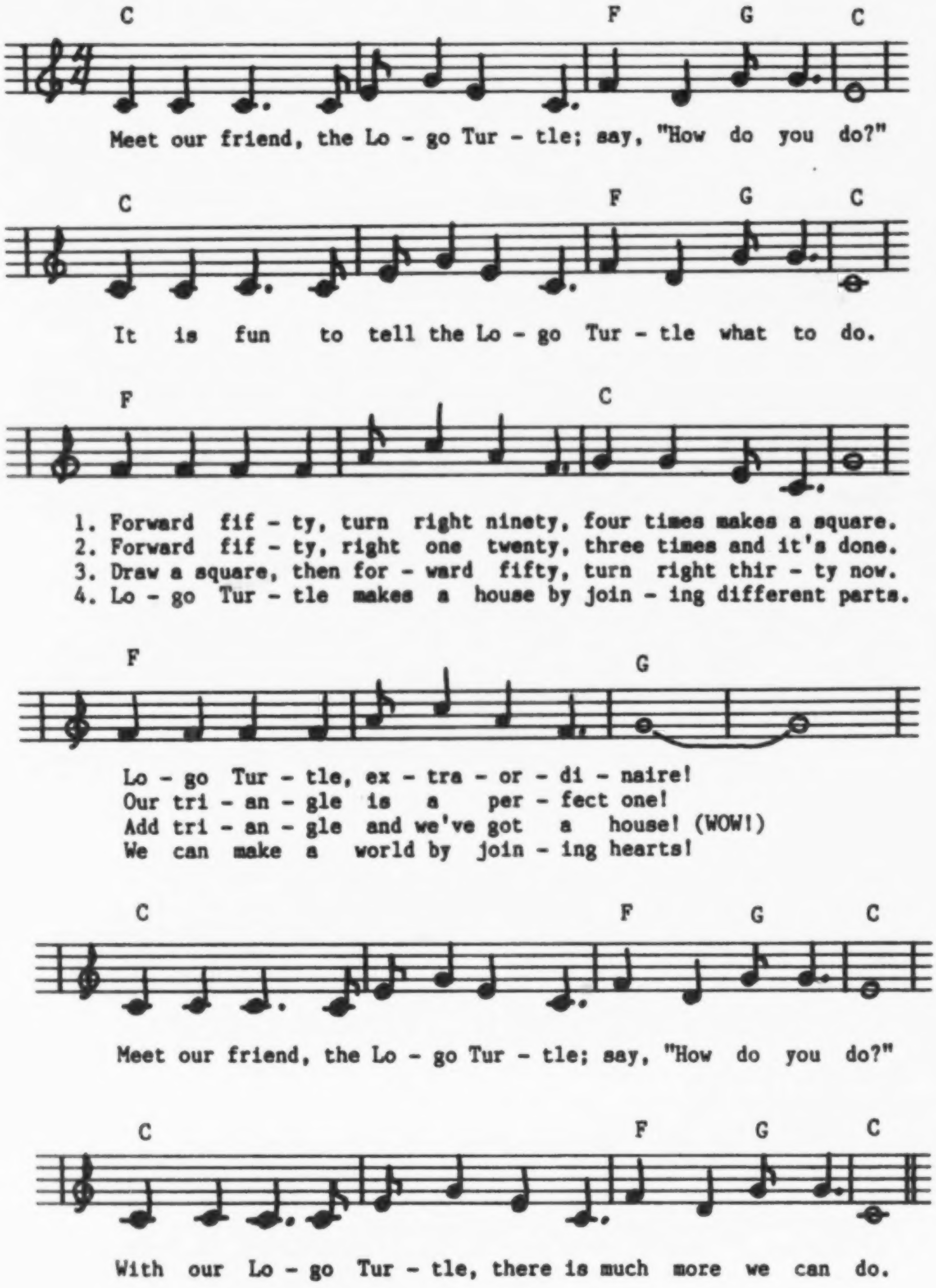
The Oakland County (MI) Logo Users Group functions as a forum where local educators can share teaching styles, materials, and ideas for implementing Logo. Through this group, a network of Logo users and resource people have been identified. The group has started a list of resource materials, including books, magazines with Logo columns or special Logo issues, articles, and manuals.

Future meetings will include an update of local district Logo activities, especially the district curriculum writing projects. These projects focus on either how to integrate Logo into existing literacy classes as a way of introducing students to the Logo language and environment, or how to integrate Logo into the existing curricular areas for use as a tool in subject areas and a means of problem solving in general,

For more information on the Oakland County Logo Users Group, contact Lynn Lieberman, Oakland Schools, 2100 Pontiac Lake Road, Pontiac, MI 48054, or call (313) 858-1896.

Turtle Songs

Words and Music by Tom Lough


C F G C
Meet our friend, the Lo - go Tur - tle; say, "How do you do?"
C F G C
It is fun to tell the Lo - go Tur - tle what to do.
F C
1. Forward fif - ty, turn right ninety, four times makes a square.
2. Forward fif - ty, right one twenty, three times and it's done.
3. Draw a square, then for - ward fifty, turn right thir - ty now.
4. Lo - go Tur - tle makes a house by join - ing different parts.
F G
Lo - go Tur - tle, ex - tra - or - di - naire!
Our tri - an - gle is a per - fect one!
Add tri - an - gle and we've got a house! (WOW!)
We can make a world by join - ing hearts!
C F G C
Meet our friend, the Lo - go Tur - tle; say, "How do you do?"
C F G C
With our Lo - go Tur - tle, there is much more we can do.

Copyright (c) 1984 Tom Lough. All rights reserved, with the exception that permission is given to photocopy this particular page of The National Logo Exchange newsletter for classroom use. The National Logo Exchange, PO Box 5341, Charlottesville, VA 22905 USA.

NLXual Challenges

by
Robs Muir

An XLent Problem

OK. Let's face it. If we were still using Roman numerals, these funny machines we call computers would be very much different, right? Can you imagine the internal architecture? A designer's nightmare! And you thought decimal to hexidecimal conversion was difficult!

AD ROMANUM

Return with us to the thrilling days of Nero et Brutus and relive the wonders of the Roman numbering schemes. Perhaps you will discover why the ancient Romans had nothing in their future; indeed, the zero had to be invented by the Persians.

As you may remember from your 5th grade math book, the Roman numerals are composed of the "numbers" I, V, X, L, C, D, M, and so on, with values represented in our (Arabic) system as 1, 5, 10, 50, 100, 500, 1000, and so on. Using a unique notation, some numbers appear to subtract from one onother. For example, IV is 4. Others appear to be cumulative; VII is 7.

NUMERI LOGORUM

This month's problem (come back here!) requires you to use your ancient QWERTY keyboard to compose a Logo program to translate ancient Roman numbers into the slightly less ancient Arabic numbers. The ultimate goal is to permit your Logo computer to add two Roman numbers together and provide the answer in our more familiar numbering system. For example,

```
ADD "MCMLXXXIV "XVI
2000
```

Of course, with some initiative, this problem could be extended to include the full range of math functions. (How would you work with fractions?) You also could have Logo return the answers in Roman numerals, and work in a Latin lesson to boot.

```
ADDERE "XVII "IX
XXVI
```

While you're at it, try some things like ABANTE "C (a.k.a. FORWARD 100, natch!).

To receive a potpourri of solutions from other readers, send your procedure listing (or your bug-infested attempts) along with a self addressed stamped envelope to:

NLXual Challenges
Attn: Robs Muir
1688 Denver Avenue
Claremont, CA 91711

Robs Muir is a teacher in the Claremont CA Unified School District and Claremont Graduate School. He presents Logo workshops in the southern California area.

Logo Some Z's is the newsletter of the Logo Users Club of Japan. Edited by Setsuko Abe and packed with information on the Japanese Logo scene, the quarterly publication costs $30 US per year, and is published in Japanese. Logo Some Z's, Geodesic, Inc., Parkheight #402, 1091 Kumagawa, Fussa-shi, Tokyo Japan.

The National Logo Exchange, copyright (c) 1984 by Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly, September through May, $25 (US) per year, mailed first class from Charlottesville, VA. $5 (US) additional per year for addresses outside the USA, Canada, and Mexico. The opinions expressed by the authors are not necessarily those of The National Logo Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress Street, Salem, MA 01971. Address other correspondence to: The National Logo Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor....................................Tom Lough

THE NATIONAL
LOGO
EXCHANGE

Volume 3 Number 4 FORWARD 100! December 1984

Functions of OUTPUT

by
Nancy Kovatch

"How can I use Logo to make algebra more meaningful to my students?" This question was my motivation for developing the Logo ideas which follow.

Although my algebra students were able to deal with linear functions in the x- and y-notation, invariably they had difficulty dealing with the function notation. The expression f(x)=2x+10 didn't come as easily as y=2x+10. When chalkboard diagrams of function machines with input and output failed to help, I realized that the Logo OUTPUT primitive might do the trick.

Here is a set of typical procedures which I typed into the computer before class.

```
TO F :X
OUTPUT 2 * :X + 10
END

TO G :X
OUTPUT :X + 2
END
```

During class, my students evaluated the functions at several points, and experimented with function operations.

```
PRINT F 3
16
PRINT (F 3) + (G 5)
23
PRINT F (G -2)
10
```

Although these functions are linear, you could use the same technique to explore quadratic, cubic, and trigonometric relationships as well.

HOUSEKEEPING

Here are some additional procedures I developed for housekeeping and to facilitate exploration.

```
TO AXES :YSIZE :XSIZE
PU HOME CS PD
Y.AXIS :YSIZE
RT 90
X.AXIS :XSIZE
END

TO Y.AXIS :Y
FD :Y BK 2*:Y FD :Y
END

TO X.AXIS :X
FD : BK 2*:X FD :X
END

TO SET.TURTLE :X :Y    (LCSI versions)
PU SETPOS LIST :X :Y PD
END

TO SET.TURTLE :X :Y    (MIT versions)
PU SETXY :X :Y PD
END
```

After drawing the AXES, my students used SET.TURTLE and the predefined functions for their explorations.

```
SET.TURTLE -50 (F -50)
SET.TURTLE 10 (F 10)
SET.TURTLE 50 (F 50)
```

Sometimes, they used this sequence.

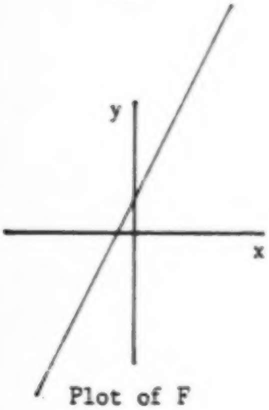
```
PRINT F 40
90
SETPOS [40 90] or SETXY 40 90
```

Once they had a good understanding of the behavior of the function, they had the turtle draw the line.

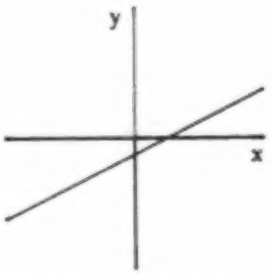
Functions continued

Another project they enjoyed was the investigation of the behavior of inverse functions. That is, if f(x)=2x+10, then the inverse function is x=(f(x)-10)/2. The inverse of the procedure F could be written

```
TO F.INV :X
OUTPUT (:X-10)/2
END
```



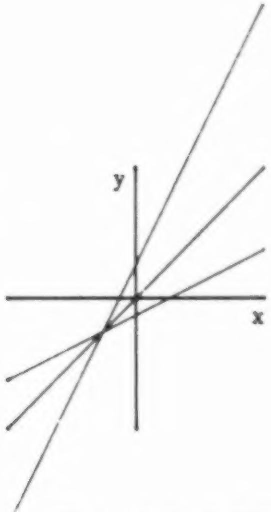
Plot of F



Plot of F.INV

When working with inverse functions, the identity line is useful. This is a line from the lower left to the upper right so that x=y at all points.

Here is what F, F.INV, and the identity line look like all together. The students fold the graphics printout along the identity line to verify the reflexive property of inverse functions.



Plot of F, F.INV, and the identity line

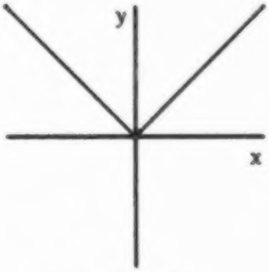
Several students experimented with ways to estimate the y-intercept. They moved the turtle along either the function line or the y-axis until the turtle was near the intercept. Then, with PRINT POS (or PRINT XCOR PRINT YCOR) and instructions to print various values of F or F.INV, they zeroed in on the intercept point.

The slope of a linear function could be investigated in a similar manner, using SETHEADING and HEADING.

WEIRDO FUNCTIONS

Many students became fascinated with "strange" linear functions. For example, the absolute value function is easy to program in Logo, but difficult for some students to understand. Exploration with the SET.TURTLE procedure gives them a personal encounter with the "beast."

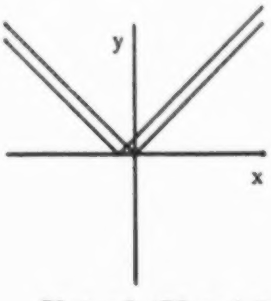
```
TO ABS :X
IF :X < 0 [OUTPUT -:X]
OUTPUT :X
END
(MIT Logo: omit the brackets above.)
```



Plot of ABS

Later, just for fun, I gave them related functions to compare, such as

```
TO B :X
OUTPUT ABS :X + 5
END
```



Plot of ABS and B

December 1984

NLXTRA

前 100

by
Hillel
Weintraub

Last month, I promised a review of Epson's Mighty Logo, but it arrived too late for me to write about this month. It is still "on the stack," however.

Instead, let me tell you about another Logo version I have been using. It is for the Epson PX-8, a portable computer being marketed in the USA and Europe under the name Geneva. It has been great fun to sit on the train and write programs in Logo!

This is a prototype version, and Epson is still considering various applications for it. If you have any ideas for capabilities you would like to see incorporated into a portable Logo, drop me a line, and I'll pass your suggestions along to Epson. They would be interested in knowing what kind of applications NLX readers envision for such a system.

LOGO VERSUS THE JAPANESE EDUCATIONAL SYSTEM

Let me tell you something of Logo's future in Japan and the general direction of the educational use of computers here. This past summer, while visiting the USA, I talked with Prof. Papert about Japan and Logo. He commented that, in France, the national educational committee had just pronounced Logo the preferred language for primary school children. I got the impression that this didn't just happen overnight, but, rather, was accomplished after a lot of behind-the-scenes work by a lot of dedicated people.

Prof. Papert felt that Japan and France had much in common, due to the centralized educational bureaucracy (called Mombusho in Japan). The kind of ground work done in France, he said, was also necessary in Japan in order to help the Mombusho see the value of using computers in ways other than just drill and practice. He felt that the Logo community should join forces with other like-minded groups of people interested in the ideas of Montessori, Piaget, Dewey, or progressive education in general. That community would then grow large enough and influential enough to affect public education decisions.

After I returned to Japan, I talked about Prof. Papert's idea with Hiroyoshi Goto, of the BYNAS Division of the UNY Company. More than anyone else in Japan, I believe, Goto-san has been working to bring Logo to the attention of the public, especially educators. He agreed with Prof. Papert's idea; in fact, BYNAS has just opened an office in Tokyo (their home office is in Nagoya). One of their main purposes is to help distribute their Logo Journal and disseminate information about Logo.

Living in Japan will do nothing if not teach one patience. That is just what it will take, along with a lot of hard work, to see any real inroads in the very conservative approach to education which is predominant in Japan. Even more than the Mombusho, the university entrance examination system is a force which keeps educational innovation and experimentation to a minimum. This test is the sole basis upon which a student's entrance to a university is decided.

Because the "name" of the university one attends is one of the most influential factors in being selected by a company (and thus being set up in a lifetime job), the students themselves often constitute the most conservative force! "Don't give me that creative stuff; just tell me what I need to memorize for the test!"

As you can see, there is a long way to go, but that is one of the things that makes living in Japan interesting! Here, our motto is MAE 100!

(Ed. note: Send your ideas on the portable Logo to Hillel at Doshisha International High School, Tatara, Tanabe-cho, Tsuzuki-gun, Kyoto-fu 610-03 Japan.)

Hillel Weintraub is a teacher at Doshisha International High School in Kyoto-fu, Japan, and is the president of the Society for Microcomputing in Life and Education (SMILE).

Dr. Logo is getting an enthusiastic reception with European computer manufacturers, according to a recent issue of Digital Research News. The manufacturer of the Apricot computer includes a copy of Dr. Logo with each machine purchased. Two other companies have announced similar plans.

Logo Disserts:

Dissertations Dealing with Logo

by Barbara Elias

Can Logo programming serve as an effective teaching tool in elementary school mathematics? This question served as the focus of a study which investigated Logo programming as one way to facilitate the understanding of number sequences and variables.

Learning to program involves active learning and can be highly motivating. Through programming activities, children can express their abilities as well as develop problem solving skills.

Eighteen 5th grade children who had no prior programming experience participated in the fifteen week study. The first five weeks were spent in learning the basics of the Logo language.

The next five weeks were devoted to learning to write procedures. During this time, the children were grouped on the basis of ability and randomly assigned to one of three instructional groups. Each group was given programming tasks along with varying amounts of information for completing each task: a complete sequence of steps to carry out, an incomplete computer program, or no additional information.

The last five week period was spent on programming tasks without any information other than the task itself.

Results indicate that programming can be a useful teaching tool in elementary mathematics. Children in the experimental group were able to write programs successfully, and demonstrated an ability to use variables. From a practical standpoint, the programming experiences facilitated the development of problem solving behaviors and served to "turn on" some children who seemed less motivated in the regular classroom environment.

>>>Milner, S. D. (1973). The Effects of Teaching Computer Programming on Performance in Mathematics (Doctoral dissertation, University of Pittsburgh, 1972). Dissertation Abstracts International, 38/08, 4183-A.

(Ed. note: Each month, Barbara Elias highlights a dissertation or thesis dealing with Logo. She is accepting copies of recent research results for the NLX Dissertation and Thesis Repository. Reports may be mailed to her in care of NLX, PO Box 5341, Charlottesville, VA 22905.)

Barbara Elias is an assistant professor in the Education Department of Virginia State University in Petersburg, VA, and a doctoral candidate at the University of Virginia.

PRINT-IT! is a plug-in card which provides the Apple computer with a Print Screen capability similar to that of the IBM PC. This allows you to make a printer copy of any screen display, whether it is graphics, text, or both. All you have to do is push a button, then press <return>.

Working with both serial and parallel printers of several dozen models, the PRINT-IT! card allows the selection of several printing options, including both high and low resolution, page 1 and page 2 side by side, and expanded, inverted, or rotated images.

The PRINT-IT! card comes with an external button, a printer cable, and a concise clearly written 25-page manual. The price is $199. For more information, write Texprint, 220 Reservoir Street, Needham Heights, MA 02194, or call (617) 449-5808.

NLX subscriber and author Sharon Burrowes reports that teachers in her school system are having difficulty with Apple Logo and the Apple IIc computers. It seems that Apple Logo disks purchased more than a year or two ago do not boot on the IIc.

If you are in this situation and desire to run your Apple Logo on a IIc, send your old disk with $15 to Logo Computer Systems, Inc. (LCSI), and they will send you a replacement which will boot.

By the way, LCSI has a new address and telephone number. LCSI, 555 West 57th Street, Suite 1236, New York, NY 10019. (212) 765-4780.

Krell Software Corp. announces its Waterloo Logo for the IBM PC. The package includes a 3-disk set and documentation for $99.95. For more information, write Krell Software Corp., 1320 Stony Brook Road, Stony Brook, NY 11790, or call (800) 245-7355.

Functions continued

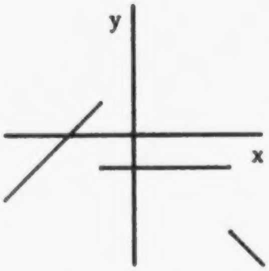
MORE WEIRDO FUNCTIONS

Here is an "exotic" linear function which gave my students quite a challenge.

For $x <= -10$ $f(x)=x+20$
For $-10<x<30$ $f(x)=-10$
For $x>=30$ $f(x)=-x$

This is the Logo procedure.

```
TO H :X
IF (OR :X<-10 :X=-10 ) [OUTPUT :X+20]
IF (AND :X>-10 :X<30 ) [OUTPUT -10]
OUTPUT -:X
END
(MIT Logo: omit the brackets above.)
```



Plot of H

By this time, the students were enthusiastically making up their own functions and challenging each other to sketch them. Logo proved to be an able arbiter of many heated discussions!

With these few beginning ideas and procedures, my students explored many second year algebra concepts in a way never before possible. Further, they were able to create, test, and modify their own predictions in an encouraging atmosphere.

The OUTPUT operation produced an output for my class: that of experienced mathematical thinkers and enthusiastic problem solvers. It can do the same for you!

Nancy Kovatch is a teacher and computer coordinator at St. Anne's Belfield School and an adjunct faculty member at Piedmont Virginia Community College in Charlottesville, VA. She is currently studying applications of Logo in second year algebra classes.

From the Editor

Looking and Seeing

"Look, Jane, look and see." What memories that phrase brought back to me! Dim visions of my wonderful first grade teacher, Miss Eloise Miller, flooded my mind as I thumbed through the worn pages of the classic primary book.

I began thinking about "look and see." I wondered why the author chose that phrase. Look and see.

My memory recreated the first class of a photography course taken years ago. The instructor was speaking.

"...and your first homework assignment is to learn how to see. You each are to select a vantage point and remain one full hour, doing nothing but looking quietly at what is there. Record your observations. After a while, start thinking about what you are looking at. Try to imagine things you could photograph. Next, imagine what small changes you could make to improve your imaginary photographs. Record these observations. At first, you will be looking. Later, you will be seeing."

Look and see. What's the difference? Thinking, imagining, ... making something of what you are looking at?

This simple but important concept has great potential for Logo. There are things happening all around us which make wonderful Logo explorations. But we have to think while we are looking at them in order to see them.

In the review section of this issue, David Thornburg's magnificent books feature explorations of patterns which can be seen in nature. The NLXual Challenge is based on an observation of Kyser as he toddled toward his shadow. Many more suggestions come if we look and see more often.

Why not encourage students to try this idea? Make Logo part of a homework assignment. What things can students find in their daily lives which could be explored with Logo?

But, be prepared. The results may astonish you!

May you see a special joy during the holiday season!

FD 100!

Tipps for Teachers

by Steve Tipps

Numbers in Between

Last month, the topic was development of a number guessing game. TEST was used to determine whether the guess was right or not. The game responded whether the guess was higher or lower than the target. However, the game did not tell the players if they were getting close to the right number.

The idea of getting close to a number may lead to another game. An improvement of the guessing game would tell when the guess is within five of the right answer. One way of expressing this would be: "Subtract the guess from the number and TEST the difference."

This simple idea can be a real mind bender because the difference can be 10, 5, 3, 1, -5, -20, and the like. You really want to tell if the guess is five too high or five too low.

How do you TEST for a differnce less than 5 but not less than -5? One way is to use ALLOF in the TEST.

```
TEST ALLOF (:NUMBER-77<5)(:NUMBER-77>-5)
IFTRUE PRINT [PRETTY CLOSE.]
```

This is read: "Find out if the difference is less than 5 <u>and</u> more than -5. If that's true, you are getting close."

<u>BETWIXT AND BETWEEN</u>

The same high/low idea could be used in a starter game of BETWEEN?. The guesser would give a low number and a high number. The game would tell whether the target number was between them.

```
TO BETWEEN? :HIGH :LOW
TEST ALLOF (:LOW < 54 ) ( 54 < :HIGH )
IFTRUE PRINT [ IT IS BETWEEN ]
IFFALSE PRINT [ IT IS NOT BETWEEN ]
PRINT1 :LOW PRINT1 [' AND '] PRINT :HIGH
END
```

The guessing game could begin by finding the range of high and low numbers. Then use GUESS.IT to get it exactly right.

```
BETWEEN? 10 40
IT IS NOT BETWEEN
10 AND 40
BETWEEN? 40 70
IT IS BETWEEN
40 AND 70
BETWEEN? 50 60
IT IS BETWEEN
50 AND 60
BETWEEN? 53 55
IT IS BETWEEN
53 AND 55
```

After narrowing the range, GUESS.IT could be used to pinpoint the number. Of course, the value of "IT needs to be the same for both the BETWEEN? and the GUESS.IT procedures. The target in BETWEEN? must be changed from a fixed number to a variable IT. The value of IT is used instead of the fixed numbers in the TEST.

<u>LINE OF NUMBERS</u>

Showing the range of numbers on the screen might be a way to help students picture which numbers are too high and which are too low. The turtle could mark the guesses or show the high and low numbers.

In the SPLITSCREEN mode, draw a number line and label it. The number line is drawn at the bottom of the graphics screen. The number labels appear on the TEXTSCREEN just under the line. Pseudocode for the number line might be:

```
1. Draw a number line using graphics
   and text.
2. Move the turtle to the bottom of
   the split graphics screen.
3. The screen is about 280 wide. I
   want 100 numbers on it. I can use
   280/10 = 28 turtle steps for each
   10 numbers.
4. Draw a line with marks every 28
   turtle steps.
5. Put 10 20 30 ... on the text line.
6. Move the cursor to line 22 to work
   out problems.
```

Three procedures are needed to make and label the number line.

```
TO NUMBER.LINE
DRAW HIDETURTLE
PU BK 75 RT 90 BK 140 PD
REPEAT 10 [MARK FD 28]
BK 1 MARK
PRINT.LABEL
END
```

Tipps continued

```
TO MARK
LT 90
FD 2 BK 2
RT 90
END

TO PRINT.LABEL
CURSOR 0 20 PRINT [0]
CURSOR 2 20 PRINT [10]
CURSOR 6 20 PRINT [20]
CURSOR 10 20 PRINT [30]
CURSOR 14 20 PRINT [40]
CURSOR 18 20 PRINT [50]
CURSOR 22 20 PRINT [60]
CURSOR 26 20 PRINT [70]
CURSOR 30 20 PRINT [80]
CURSOR 34 20 PRINT [90]
CURSOR 37 20 PRINT [100]
CURSOR 0 22 PRINT [] CURSOR 0 22
END
```

The number line can be used in many ways. Guesses can be put on it. For example, each time a number is guessed, a STAR is drawn at that number on the line.

HOME ON THE RANGE

Small stars or marks may be hard to read on the screen. A better way would be to have the range marked on the number line. A RANGE procedure draws a red line from 0 to the :LOW number, then a green line from the :LOW to the :HIGH, and finally a red line from :HIGH to 100 on the number line.

```
TO RANGE :LOW :HIGH
HT PU HOME
BK 40 RT 90 BK 140 PD
PC 4 FD :LOW / 10 * 28
PC 2 FD (:HIGH - :LOW) / 10 * 28
PC 4 FD (100 - :HIGH) / 10 * 28
END
```

Try out the number line and RANGE several times to see how they work. Get a clear picture of how many numbers are between the low and high numbers.

RETURN OF BETWEEN

The old BETWEEN procedure can be elaborated upon with the visual display in RANGE. The IS.IT.BETWEEN? procedure has more information and more feedback about the range of numbers. One refinement is a filter at the very beginning to check for :LOW being lower than :HIGH.

```
TO IS.IT.BETWEEN?  :LOW :HIGH
FILTER  :LOW :HIGH
TEST ALLOF (:LOW < :IT)(:IT < :HIGH)
IFTRUE PRINT [ YES, IT IS.]
IFFALSE PRINT [ NO, IT IS NOT.]
IFTRUE RANGE :LOW :HIGH
DELAY 3000
PRINT.LABEL
END

TO FILTER :LOW :HIGH
TEST :LOW < :HIGH
IFFALSE  PRINT  [ YOU  ARE  TRYING  TO
TRICK ME.]
IFFALSE PRINT [TRY AGAIN.]
IFFALSE TOPLEVEL
END


TO DELAY :NUMBER
REPEAT :NUMBER []
END
```

A START procedure calls all the other procedures necessary to begin the game of IS.IT.BETWEEN? The RANDOM.SET.IT procedure could assign a random number as the target. For ideas on how to do that, see Patrick Salsich's article in this issue. A complete set of printed instructions would be a good addition.

```
TO START
RANDOM.SET.IT
INSTRUCTIONS
NUMBER.LINE
END
```

By asking IS.IT.BETWEEN? several times, you get a smaller and smaller range. If the random number was 72, the game would look like this.

```
IS.IT.BETWEEN? 0 50
NO, IT IS NOT.
IS.IT.BETWEEN? 50 100
YES, IT IS.
IS.IT.BETWEEN? 50 75
YES, IT IS.
IS.IT.BETWEEN? 65 75
YES, IT IS.
IS.IT.BETWEEN? 70 75
YES, IT IS.
```

Each time a number line is redrawn to show the green range. An exploration focuses on the range of right numbers regardless of whether the exploration is

Tipps continued

systematic or haphazard. At the end of the exploration, a direct guess may be wanted. GUESS is changed to an IS.IT? procedure which also puts a star at the right place on the number line.

```
    TO IS.IT? :NUMBER
    TEST :IT = :NUMBER
    IFTRUE PUT.STAR :NUMBER
    PRINT.LABEL
    IFTRUE PRINT1 :NUMBER PRINT1 [' IS THE
 RIGHT NUMBER.']
    IFFALSE PRINT1 :NUMBER PRINT1 [' IS
NOT THE NUMBER.']
    END

    TO PUT.STAR :NUMBER
    HT PU HOME
    BK 40 RT 90 BK 140 PD
    PC 4 FD :NUMBER/ 10 * 28
    PC 3 LT 90 FD 35
    PC 2 LT 15 STAR RT 15
    PC 3 BK 50 FD 15 RT 90
    PC 4 FD (100 - :NUMBER )/10 * 28
    END

    TO STAR
    REPEAT 5 [ FD 30 RT 144]
    END
```

Only three or four procedures are needed to create the range game. But each one of the procedures should challenge you and the students to think about the position of numbers and how to show that position with Logo. The modular structure of Logo is a great aid in solving this problem. Each part can be worked out, defined, debugged, and combined.

The procedures for guessing and estimating and playing with numbers should look very different by the time students have worked on them. GUESS and BETWEEN? are starters because they open up many other possibilities.

Working out how things look on the screen may be as challenging a problem as the TEST and consequences. But whether the number is too high, too low, in between, or just right, Logo is a marvelous tool for logical thinking and problem solving.

Steve Tipps is the West Professor of Education at Midwestern State University in Wichita Falls, TX. He presents Logo workshops for school systems throughout the United States.

NLXionary

A Lectionary of Selected Logo Readings

by Griff Wigley

The following selection is from a paper written for Logo 84, the National Logo Conference held at MIT this past June, and is included in the preproceedings.

(Ed. note: To obtain a copy of the Logo 84 preproceedings, send $15 US to MIT Logo Group / Publications, NE 43-438, 545 Technology Square, CAmbridge, MA 02139.)

"Logo as an Empirical Window," by Sylvia Weir (an excerpt from her forthcoming book, "Untrapping Intelligence").

>Although I don't know her personally, Sylvia Weir seems to be the Salvador Minuchin of Logo teachers and researchers. Family therapists amongst you, listen up. Therapist A asks the family what the problem is, and then suggests a solution based on what they say. When this doesn't work, Therapist A typically ponders, "How can I make my suggestion clearer?"

>Therapist B asks the family to demonstrate the problem and then prescribes a task that seeks both to change and further diagnose the structure (pattern of interaction) of the family. When this doesn't work, Therapist B typically ponders, "What else could I try to change the structure so that they can solve this and future problems themselves?"

>Yes, therapists as well as teachers frequently have their own agenda to impart to their clients, and tend to strive for greater clarity when hitting a brick wall. (Need I mention blaming as an alternate response?)

>Sylvia Weir and Salvador Minuchin go deeper. In this excerpt, Weir challenges teachers to ask, "How can I set up situations so as to enable my students to find their way to an understanding?" Likewise, she challenges researchers to ask, "What must I provide so that what my subjects know already, as well as what they can come to know, will surface in a clear way for me to observe and probe?"

>What could possibly be beneficial to observing and analyzing something so insignificant as family members seating them-

NLXionary continued

selves for a session, or a student's first attempt at choosing a series of numbers to go with an initial exploration of FD and BK? The true professionals among you will want to know.

>"Logo as a window" is a phrase that evokes memories of the one-way glass and videotape of my family therapy training. Few therapists really use them as diagnostic tools. They take too much time.

>"Got to get on with solving the problem." "These students should be learning REPEAT by now." Curriculum, curriculum, curriculum.

>Like most of the really good stuff about Logo, its potential as a window is available only through the skill of a teacher. I leave it to you to decide whether that is good news or bad news. In either case, watch for her book!

"SmartLogo," a review by David Wilson, InfoWorld, November 12, 1984.

>A glowing 4-disk rating review of LCSI's Logo for the Coleco Adam home computer. Among the positives: a speedy, turtle-like turtle; 16 colors for ink or screen; 30 turtles with a shape editor; commands for fill, speed, sound, and music; great documentation; and "more list processing capabilities than we have ever seen in any prior version of Logo for microcomputers." The only fault the reviewer finds is with the computer - no disk software (the Adam uses a tape drive), and no graphics printer.

>The reviewer writes as if he has never seen any of these capabilities in Logo versions before, and mentions other versions only to compare turtle drawing speed. He says SETSPEED is a totally new primitive when, in fact, it has been around since 1981 when TI Logo was released.

>Actually, none of the features he mentions are brand spanking new. (However, SmartLogo does sound like a top contender with LCSI's Sprite Logo for the most complete Logo version to date.) Clearly, a more thorough review of SmartLogo is needed quickly, or the Adam may never make it out of the Cabbage Patch. (And send the man some NLX back issues!)

Griff Wigley is a co-owner of Family Computing, Inc., a facilitator in the Faribault (MN) Public Schools, and a school board member of Prairie Creek Community School.

TurtleTips

by
Donna Lanyi
and Jane Toth

So how do we get to Michael Jackson's house? Students may need additional practice in guiding the turtle. An excellent gamelike activity to practice this concept makes use of mazes.

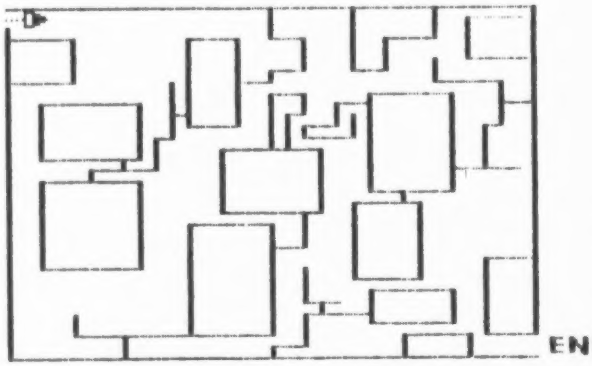
TEACHER HINTS

We trace commercially prepared mazes onto transparency material. There are some excellent mazes in the MECC Logo publications. (For a current catalog, write MECC, 3490 Lexington Avenue North, St. Paul, MN 55112.) The students tape the mazes to their monitors and practice guiding the turtle through.

If the transparency material is light enough, static electricity will hold it to the screen. Turning the monitor off and on sometimes seems to make more static "cling."

Encourage your students to create their own mazes, such as how to get to MJ's house, to Santa's Workshop, and the like. They really enjoy practicing with mazes. Our students think of the activity more as a video game than turtle practice!

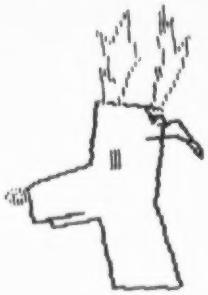
Thanks to Elizabeth Moody, former Computer Coordinator at Trinity School in New York City, who sent us a procedure to draw the following maze her students used for practice.



TurtleTips continued

PROCEDURE OF THE MONTH

Happy Holidays!! Encourage your students to create graphics for the holiday season. How about a picture of one of Santa's reindeer? Or all eight?



If you are interested in a free listing of this procedure or Elizabeth's maze procedure, please send us a self addressed stamped envelope.

We invite elementary teachers and students to send their procedures or ideas to be shared. Donna Lanyi and Jane Toth, Beall Avenue School, 716 Beall Avenue, Wooster, OH 44691.

Donna Lanyi and Jane Toth use Logo in their elementary classrooms in the Wooster (OH) City Schools.

Really Random

by

Patrick Salsich

(Ed. note: A similar version of this article appears in the December 1984/January 1985 issue of The Computing Teacher by special arrangement.)

Is RANDOM really random? Recently, I provided children with a procedure which used RANDOM and RANDOMIZE to place an object on the screen in a random position. They turned on their five Commodore 64 computers, booted Commodore Logo, loaded the procedure, ran it, and got the same results in the same sequence. The effect was spoiled.

As a result, I decided to try to add some kind of unpredictability to the procedure. Here is what I learned.

A LOOK AT RANDOM

Have you ever tried to generate a random number with Commodore Logo? It is not as easy as it seems. RANDOM is supposed to output a random number within certain limits, but, in fact, it outputs the first number in a predetermined sequence designed to simulate randomness. This approach is standard with many computers.

The RANDOMIZE command is used to select the starting point in the sequence RANDOM uses. Unfortunately, if RANDOMIZE is used with no input, it starts at the same number in the sequence each time the computer is turned on. Try the following to see what I mean.

Turn on your Commodore 64 and boot Commodore Logo. Then define and run the following procedure.

```
TO TRY.RANDOM
RANDOMIZE
REPEAT 10 [PRINT RANDOM 5]
END
```

Write down the sequence of numbers generated. Now turn the computer off, then on, and reboot Logo. Type and run the same procedure. The same sequence of numbers is generated.

We can change the sequence. RANDOMIZE can accept an input which determines the point (called the "seed") in the random number sequence at which RANDOM begins. For example, if you replace RANDOMIZE with (RANDOMIZE 56) in the TRY.RANDOM procedure (the parentheses are required), a different sequence is produced, but the sequence of numbers does not change each time the procedure is run.

If we could arbitrarily choose the input for RANDOMIZE, we could produce an unpredictable result with TRY.RANDOM each time the computer is turned on. Here is one approach to the task.

HOW TO RANDOMIZE

GROW.NUMBER is a procedure which counts up from an initial number until it is interrupted with a key press. Run it a couple of times by typing GROW.NUMBER 0 to verify that it is difficult to produce the same number more than once.

```
TO GROW.NUMBER :COUNTER
IF RC? PRINT :COUNTER STOP
GROW.NUMBER :COUNTER + 1
END
```

Random continued

Now change the procedure so that it uses the stopping value of COUNTER as the input to RANDOMIZE.

```
TO GROW.NUMBER :COUNTER
IF RC? (RANDOMIZE :COUNTER) STOP
GROW.NUMBER :COUNTER + 1
END
```

With the addition of a prompting procedure and an appropriately changed TRY.RANDOM procedure, we're in business.

```
TO GET.RANDOM.SEED
PRINT [PRESS ANY KEY TO BEGIN]
GROW.NUMBER 0
TRY.RANDOM
END

TO TRY.RANDOM
REPEAT 10 [PRINT RANDOM 5]
END
```

When I used a similar set of procedures with the children later, no one got the same results, even when each began with booting up. Thus, this is one way of generating more unpredictable sequences of numbers with Commodore Logo.

(Ed note: To eliminate the spurious character on the screen after GROW.NUMBER stops, change the appropriate line to IF RC? MAKE "KEY RC (RANDOMIZE :COUNTER) STOP. In case you are interested, RANDOMIZE accepts as input a number less than 2.147485E9.)

Patrick Salsich is an elementary teacher on special assignment in the Portland (OR) Public Schools to develop a program of Logo instruction for grades 1-5. He is pursuing a Masters Degree in Computer Science Education at the University of Oregon, and is the founder of the Portland Area Logo Users' Group.

Logo Templates for Apple

Keyboard templates which display Logo commands are available for the Apple II+ and IIe computers. A student model includes an attached booklet with pages which allow the insertion of lessons or assignments. An advanced model displays a wide variety of Logo commands.

For more information, write Integrated Computer Marketplace, Box 3610-I, SR 2, Tehachapi, CA 93561, or call (805) 822-7575.

Listful Thinking

by

Glen Bull and Paula Cochran

Making a List and Checking It Twice

It's that time again! Perhaps you and your students already have some holiday lists in mind -- things to do, things to buy, friends to remember. This month, we'd like to show you how to check your lists. We also have two holiday games for you.

MAKING A LIST

Next to that of the seven dwarfs, the list of the names of Santa's eight reindeer is among the most frequently forgotten lists of all time. How many can you remember? We'll start you off with a list of two reindeer.

```
MAKE "REINDEER [COMET DANCER]
```

As you recall from last month, INSPECT is a good way to look at a list. It prints one reindeer on each line. In Terrapin Logo, omit the brackets around [STOP].

```
TO INSPECT :LIST
IF :LIST = [] [STOP]
PRINT FIRST :LIST
INSPECT BUTFIRST :LIST
END
```

Here is your team so far, Santa.

```
INSPECT :REINDEER
COMET
DANCER
```

Last month, we made a list of pilgrims and created a short procedure to ADD more pilgrims to the list. This month, we can use this tool to ADD reindeer.

```
TO ADD :ITEM :LIST
MAKE :LIST LPUT :ITEM THING :LIST
END

ADD "CUPID "REINDEER
INSPECT :REINDEER
COMET
DANCER
CUPID
```

Listful Thinking continued

GUESS THE REINDEER

The tools INSPECT and ADD make a reindeer guessing game. If your Logo version does not have a READWORD primitive, a READWORD procedure is furnished at the end of the column.

```
TO REINDEER.CHALLENGE
INSPECT :REINDEER
PRINT [NAME A NEW REINDEER.]
MAKE "NAME READWORD
ADD :NAME "REINDEER
END
```

Oops! On inspecting the reindeer, it appears that someone added DANCER twice.

```
INSPECT :REINDEER
COMET
DANCER
BLITZEN
DANCER
```

No problem! You can use the EDNS (EDIT NAMES in Terrapin Logo) command to edit the list. This method for editing lists was discussed in last month's column.

CHECKING THE LIST

But, why not let Logo check a reindeer before you add it to the list? Here is a way to find out whether a reindeer is already a member of Santa's team.

```
PRINT MEMBERP "DANCER :REINDEER
TRUE
PRINT MEMBERP "RUDOLPH :REINDEER
FALSE
```

***In Version 1 of Terrapin Logo, you need to use the MEMBER? procedure found at the end of the column. In Version 2 of Terrapin Logo, the MEMBER? command is built in.

The MEMBERP command is used to check the list before adding a reindeer:

```
TEST MEMBERP :NAME :REINDEER.
```

This line doesn't do anything by itself, but we can use it to improve the REINDEER.CHALLENGE.

```
TO REINDEER.CHALLENGE
PRINT [NAME A MISSING REINDEER.]
MAKE "NAME READWORD
TEST MEMBERP :NAME :REINDEER
IFTRUE [PRINT SE :NAME [IS ALREADY HARNESSED.]
IFFALSE [ADD :NAME "REINDEER]
END
```

Here's what happens if you try to add a reindeer that's already on the list.

```
REINDEER.CHALLENGE
NAME A MISSING REINDEER
DANCER
DANCER IS ALREADY HARNESSED.
```

THE ANAGRAM GAME

An anagram is a word made from the letters of another word. We have used Logo to make a holiday anagram game. There are two skills involved in playing this game: combining letters to make new words, and remembering what words have already been used. Here is what it's like to play:

```
ANAGRAM
THINK OF A WORD THAT ONLY USES
LETTERS FOUND IN "GREETINGS"

WHAT IS YOUR WORD?
SING   (typed by player)
THANKS! THAT IS A NEW ONE.

NOW THE WORDS ARE:
TIN
GET
TRITE
SING
```

You don't have to use all of the letters in the key word, and you may use some letters more than once. But you can't use a word that is already on the list! You must think of a new one.

```
WHAT IS YOUR WORD?
GET   (typed by player)
I ALREADY KNOW THAT ONE.
```

The ANAGRAM game starts off with instructions for the players.

```
TO ANAGRAM
MAKE "KEY.WORD "GREETINGS
MAKE "WORD.LIST []
PRINT [THINK OF A WORD THAT ONLY]
PRINT SE [USES LETTERS IN] :KEY.WORD
PRINT []
GET.GUESS
END
```

Listful Thinking continued

Now the player guesses a word.

```
     TO GET.GUESS
     PRINT [WHAT IS YOUR WORD?]
     MAKE "GUESS READWORD
     TEST MEMBERP :GUESS :WORD.LIST
     IFTRUE [PRINT [I ALREADY KNOW THAT
ONE.]]
     IFFALSE [ADD.TO.LIST]
     PRINT []
     GET.GUESS
     END
```

***In Terrapin Logo, omit brackets around consequences. For example, type

```
IFFALSE  ADD.TO.LIST  instead of
IFFALSE [ADD.TO.LIST]
```

If the player guesses a word that is already a member of the WORD.LIST, the program says, "I already know that one." If the guess is not a member of the list, the program adds it to the list.

```
TO ADD.TO.LIST
ADD :GUESS "WORD.LIST
PRINT [THANKS! THAT IS A NEW ONE.]
PRINT []
PRINT [NOW THE WORDS ARE:]
INSPECT :WORD.LIST
END
```

Use CONTROL-G to stop the ANAGRAM program.

Challenge: Can you improve ANAGRAM so the program stops when a player types QUIT?

CHECKING IT TWICE

The ANAGRAM program can remember the words you typed in already, but it has a flaw.

```
WHAT IS YOUR WORD?
CELEBRATION
THANKS! THAT IS A NEW ONE.

NOW THE WORDS ARE:
TIN
GET
SING
CELEBRATION
```

The program accepted CELEBRATION even though many of the letters in "Celebration" are not in "Greetings". In some versions of Logo, MEMBERP or MEMBER? check to see if a letter is a member of a word. Here is a table we compiled for several versions of Logo.

Name	Version	Check Letters?
MEMBERP	IBM Logo	Yes
MEMBERP	Apple Logo	No
MEMBERP	Apple Logo II	Yes
MEMBER?	Terrapin 1 Logo	Yes (Procedure)
MEMBER?	Terrapin 2 Logo	Yes
MEMBER?	Commodore Logo	Yes

Even though the MEMBERP command in Apple Logo will not check to see if a letter is a member of a word, we can write a procedure that will, just as we did for Version 1 of Terrapin Logo. We'll call it MEMBERX.

```
TO MEMBERX :ITEM :LIST
IF EMPTYP :LIST [OUTPUT "FALSE]
IF :ITEM = FIRST :LIST [OUTPUT "TRUE]
OUTPUT MEMBERX :ITEM BUTFIRST :LIST
END
```

Here's how to use MEMBERX to see if a letter is in a word using Apple Logo. If you are using IBM Logo or Apple Logo II, substitute MEMBERP. If you are using Terrapin or Commodore Logo, substitute MEMBER? in place of MEMBERX.

```
PRINT MEMBERX "N "CANDLE
TRUE
PRINT MEMBERX "V "CANDLE
FALSE
```

Here's a procedure to compare all the letters in a guess with the letters in the key word "Greetings". You don't have to understand everything about how this procedure works to use it. It follows the same principles that were used to check words in the reindeer list.

```
      TO COMPARE.LETTERS :GUESS :KEY.WORD
     IF :GUESS = " [OP "TRUE]
     TEST MEMBERX FIRST :GUESS :KEY.WORD
     IFFALSE  [PRINT (SE FIRST  :GUESS  [IS
NOT A LETTER IN] :KEY.WORD)]
     IFFALSE [OUTPUT "FALSE]
     OP COMPARE.LETTERS BF :GUESS :KEY.WORD
     END
```

***In Terrapin Logo, use MEMBER? instead of MEMBERX and omit brackets around consequences. For example, type:

```
IF :GUESS = "  OP "TRUE  instead of
IF :GUESS = "  [OP "TRUE]
```

Listful Thinking continued

Here's how to use COMPARE.LETTERS to check all the letters in a guess.

```
PRINT COMPARE.LETTERS "GRIPE "GREETINGS
P IS NOT A LETTER IN GREETINGS
FALSE

PRINT COMPARE.LETTERS "TIN "GREETINGS
TRUE
```

Now add instructions to GET.GUESS to check the letters in each guess with COMPARE.LETTERS. If the letters are OK, the GET.GUESS procedure goes on to see if the guess is already on the word list.

```
TO GET.GUESS
PRINT [WHAT IS YOUR WORD?]
MAKE "GUESS READWORD
TEST COMPARE.LETTERS :GUESS :KEY.WORD
IFFALSE [STOP]
TEST MEMBERP :GUESS :WORD.LIST
IFTRUE [PRINT [I ALREADY KNOW THAT ONE.]
IFFALSE [ADD.TO.LIST]
PRINT []
GET.GUESS
END
```

***In Terrapin Logo, omit the brackets around consequences. For example, type:

```
IFFALSE STOP  instead of
IFFALSE [STOP]
```

If the guess is still OK after it has been checked twice, it is added to the word list. If a player tries to use illegal letters in his guess, the program stops.

We hope you enjoy making lists and checking them twice. What's on our gift list? We would like to receive some of the games you make with these list tools.

Happy holidays!

PROCEDURES

Here are Terrapin Logo procedures for MEMBER? and READWORD.

```
TO MEMBER? :ITEM :LIST
IF :LIST = [] OUTPUT "FALSE
IF :LIST = "  OUTPUT "FALSE
TEST :ITEM = FIRST :LIST
IFTRUE OUTPUT "TRUE
OUTPUT MEMBER? :ITEM BUTFIRST :LIST
END
```

```
TO READWORD
MAKE "INPUT REQUEST
IF :INPUT = [] OUTPUT "
OUTPUT FIRST :INPUT
END
```

A READWORD procedure is in the Apple Logo STARTUP file. In case it has been erased, here is a listing for Apple Logo.

```
TO READWORD
MAKE "INPUT READLIST
IF EMPTYP :INPUT [OUTPUT "  ]
OUTPUT FIRST :INPUT
END
```

Glen Bull is a professor in the University of Virginia's School of Education, and teaches Logo courses at both the graduate and undergraduate level. Paula Cochran is a reformed English major who studied linguistics at the University of Cambridge. She is now a speech-language pathologist working with language-disabled children.

Teacher to Teacher

Logo, Math, and Art Books for Grade 4 through 12

reviewed by
Lou Wichert

Books Reviewed:

Discovering Apple Logo: An Invitation to the Art and Pattern of Nature and Computer Art and Animation: A User's Guide to Commodore 64 Logo, by David Thornburg, published by Addison-Wesley.

WHAT IF ...

...problem solving and creative thinking were priorities in the classroom?

...students were encouraged in projects of their own design?

...music and math, or art and science, were taught as one subject?

David Thornburg must have lived through these experiences! His books, Discovering Apple Logo and Computer Art and Animation, give teachers new tools to

Teacher to Teacher continued

experiment with and new approaches to teaching math and art with Logo. In both books, an abundance of teaching ideas are concisely and concretely presented in a manner sure to spark the creative mind.

As we read these books, I and the other members of our review group could not resist jotting teaching ideas in the margins as they cascaded into our minds. Thornburg offers dependable directions, creative ideas, and stimulating projects to try ... a find for busy classroom teachers!

Thornburg moves the reader from the beginning stages of Logo into the more complex manipulation of the language through specific examples and challenging questions, interspersed with a great sense of humor.

SIMILARITIES

Each book contains:

*Basic beginning directions for getting started with the turtle and turtle graphics.

*Introduction of primitives, color, procedures, variables, and editing commands.

*Utility commands, such as SAVE.

*X and Y coordinates with illustrations of uses.

*Commands and operations which can be used with list processing, such as MAKE, PRINT, and SENTENCE.

*Illustrations on how to incorporate a counter into a procedure.

*Demonstrations of procedures with explanations given to help the reader understand the process.

*Numerous ideas and projects to try.

*An attractive layout, including generous margins and spacing, bold print, and clear crisp illustrations.

*An easy to read style of writing.

*Excellent resources for the Logo teacher who would like to move beyond static images.

DIFFERENCES

*After the introduction, each book moves into an area of specialization, such as geometry, art, or computer animation.

*Discovering Apple Logo has a geometry emphasis. It offers the Logo user an opportunity to make use of geometric principles while looking to nature as a guide.

*Computer Art and Animation has an emphasis on animation, not surprisingly. The computer becomes a new tool for creative expression.

UPON CLOSER SCRUTINY

Thornburg states in Discovering Apple Logo that ancient mathematics has enchanted people for a millenium. He moves into this realm with the introduction of the golden mean and the Fibonacci series. What is the pattern when given the numerals 1 1 2 3 5 8 13 21 34 55 and so on? This numerical series is based on the golden mean. This pattern appears in many natural materials, such as pine cones or sunflower seed clusters. This is a fascinating subject for the math teacher who is interested in using Logo to illustrate this powerful concept.

Tessellations (patterns made from interlocking motifs) and star formulas are illustrated and explained.

Have you ever thought about how to measure a coastline? If you stretch our a piece of string to its fullest length, it is straight, and covers a specific distance. Take that same piece of string and make it curve and bend; notice that the distance it now covers is less. We have now entered the study of fractal geometry.

A fractal is defined as a mathematical set or concrete object whose form is extremely irregular and/or fragmented at all scales. Fractal geometry and the triadic Koch curve (a famous fractal curve) are used and explained in the study of a coastline.

Randomness in geometry allows the replication of the random patterns that we see in nature. Logo is the vehicle to use with students to illustrate these patterns.

An extensive annotated bibliography is included, and contains sections on turtle geometry, polygons and polyhedra, static symmetry, dynamic symmetry, tessellations, fractals, patterns in nature, and recursion.

In Computer Art and Animation, Thornburg moves beyond the basics of Logo into the realm of computer graphics. The kaleidoscope has always been a fascinating art form. The author shows us ways to create our own Logo kaleidoscope.

Thornburg describes multiple turtles (called "sprites" in Commodore Logo) and demonstrates how to move them on the screen. The art form now becomes animation, with helpful hints on how to do the animation and preserve the stories.

Exciting and innovative ideas to use with photography and video cassette recorders and a chapter on additional drawing aids complete the text. Creative teens love producing their own films; what better way to create good community public

Teacher to Teacher continued

relations than with a video production of Logo masterpieces?

This book is the culmination of Thornburg's "playing around" with computer art and animation, and describes what he learned along the way.

The final statement in the book says it: "Above all, have fun!"

IN SUMMARY

Thornburg has provided beginning Logo directions and moved into specialized areas. He has taken some often used school projects and revitalized them for the computer with Logo. He has given teachers a "shot in the curriculum" by asking:

...what better way to illustrate geometric principles than with the computer?

...why not use the computer as an art form?

...why not combine art and math? science and music?

We hope these stimulating texts continue. As teachers use them, they themselves could become the tools for curriculum change.

Additional books by David Thornburg.

Addison-Wesley, 2725 Sand Hill Road, Menlo Park, CA 94025.

Computer Art and Animation: A User's Guide to TI/99-4A Color Logo. 1983, $12.95.

Computer Art and Animation: A User's Guide to TRS-80 Color Logo. 1983, $12.95.

Picture This! An Introduction to Computer Graphics for Kids of All Ages. 1982, $14.95.

Picture This Too! An Introduction to Computer Graphics for Kids of All Ages. 1982, $14.95.

Practice This! An Introduction to Computer Graphics for Kids of All Ages. 1982, $14.95.

Compute Publications,PO Box 5406, Greensboro, NC 27403.

Every Kid's First Book of Robots and Computers. 1982, $4.95.

Please send your comments, questions, and concerns on this review to Logo Information For Teachers (LIFT), PO Box 5396, Plymouth, MI 48170.

Lou Wichert is a media specialist in the Farmington (MI) Schools system and an educational consultant.

Q and A

by

Jim McCauley

Q. Why do some Logo people advise against the use of MAKE and global variables? I find them really useful.

<[-?-]>

A. Computer languages always offer some way of "binding" values to variable names so that the programmer can work with the name of some value rather than the value itself. This allows the same program to produce different outputs when different values are associated with the variables within it.

LETTING IT, MAKING IT

In BASIC, the programmer has only one way of binding values to variables: by using the equals sign to assign new values to variable names. For example, this line in a BASIC program:

```
100 LET A = A + 1
```

assigns a new value to the variable name "A". The new value depends on the previous value; if A is bound to 3, running this line would assign the value 4 to A, and running it again would assign 5 to A. This line is an example of a "global" assignment statement in BASIC; when the value of A is changed in this way, the new value applies to every occurrence of A in the entire program.

The MAKE command allows one to do the same thing in Logo:

```
MAKE "A :A + 1
```

has the same effect as the line of BASIC above. If the value of :A is 3, then after this line is run, :A has a new value of 4, and that new value will be true for every :A in any Logo procedure in memory.

ANOTHER WAY

There is another way of binding values to variables in Logo that BASIC and some other computer languages lack. Logo is a procedural language, and its procedures accept inputs. For example:

Q and A continued

```
TO ADD1 :NUM
PRINT (SENTENCE :NUM [PLUS 1 IS ] :NUM + 1)
END
```

This procedure accepts a number as input, binding it to the variable name :NUM, and prints a sentence about what happens if you add 1 to :NUM. You might think that this binding is the same kind as MAKE produces, but that's not the case; as soon as ADD1 finishes executing, there is no longer anything bound to :NUM.

If :NUM occurs in some other procedure in the workspace, whatever is bound to :NUM in ADD1 won't interfere with it. The binding to :NUM is local; it applies only within ADD1 and any procedures that ADD1 might call on while it is running. ADD1 contains only primitives, but if it did call on another procedure, :NUM would still carry the value passed to it when ADD1 was called, unless a new value for :NUM is "sent" to the procedure called from within ADD1.

The ability to bind values locally (instead of only globally) is one of Logo's great strengths, because it allows us to harness the power of recursion. Here's a simple tail-recursive variation on ADD1:

```
TO ADD2 :NUM
IF :NUM > 50 [STOP]
PRINT (SENTENCE :NUM [PLUS 2 IS ] :NUM + 2)
ADD2 :NUM + 1
END
```

This will add 2 to the number bound to :NUM and tail-recursively "call itself" to continue adding 2 to the sum of :NUM and 1 until :NUM is greater than 50, when it will stop. Notice that we didn't have to use MAKE to assign a new value to :NUM; passing it tail-recursively to the new copy of ADD2 took care of that automatically. Here's a somewhat trickier variation:

```
TO ADD3 :NUM
IF :NUM > 50 [OUTPUT :NUM]
OUTPUT (SE :NUM [PLUS 3 IS] ADD3 :NUM + 3)
END
```

Type this in, and then call it with PRINT ADD3 40. The result resembles something children who have just learned to add use to drive their parents crazy at the dinner table.

THE MAIN MATTER WITH MAKE

The real problem with excessive use of MAKE is that one tends to carry over old programming habits from other languages (especially BASIC), thus missing the elegance of the parameter-passing style of programming and use of recursion that is so natural to Logo.

Use of MAKE inside recursive programs is potentially dangerous; the usual consequences are mysterious bugs related to confusion of local and global bindings. In addition, MAKE statements have a voracious appetite for memory. It is not unusual for a recursive procedure with MAKE to produce an "out of memory" error message.

I like to encourage my students to bind values to variables as inputs, rather than using MAKE, because of the clarity and elegance of that style of programming. Also, if my students have learned to program another language that makes heavy use of the assignment statement (like BASIC or Pascal), I find that this helps them make a clean break from previous ideas about programming that may cripple their capacity to learn more powerful and flexible techniques in Logo.

(Ed. Note: If you have questions about Logo programming techniques or the thinking which leads to Logo programming, send them to NLX Q & A, PO Box 5341, Charlottesville, VA 22905. Each month, Jim McCauley answers selected questions in this column.)

Jim McCauley is a Coordinator of Computer Education for the Santa Clara (CA) County Office of Education, and has written Logo articles for many national publications.

NLXual Challenges

by

Robs Muir

Of Sunshine and Shadow

One of the more exciting developments in Logo-based education is the concept of a microworld - a tightly controlled environment designed to encourage exploration of a specific idea or unit of curricula. A popular Logo microworld is the dynaturtle, a turtle consciously designed to simulate an object that obeys Newton's Laws of Motion

NLXual Challenges continued

(see Andy diSessa's "Unlearning Aristotelian Physics: A Study of Knowledge-based Learning," Cognitive Science 6, 1982).

Perhaps the most well known Logo microworld is the turtle itself; turtle graphics represents a controlled environment intended to encourage discoveries of geometric and mathematic ideas. In fact, Logo could exist without the turtle microworld; the language just wouldn't be quite so graphic.

Seymour Papert calls microworlds "incubators for knowledge." It is a pity that there aren't more well-developed microworlds such as dynaturtles to use in our classrooms.

LONG AGO IN A MICROWORLD FAR AWAY

This month's problem is an attempt to create a SUNWALKER microworld. Here's the scenario...

Tom Lough, walking with his family late one afternoon, observed his son, Kyser, discover his shadow. On his short little legs, Kyser began chasing his shadow across the lawn. After finally catching him, Tom began to wonder what Kyser's path would look like if he were allowed to follow his shadow from sunup to sundown, and suggested this problem as a result.

Since the sun's position changes during the day, the direction of Kyser's shadow would also change. Can you intuit Kyser's path? Now write a Logo program that simulates our SUNWALKER's path.

Here's a possible general method. Set the turtle in the upper right corner of the screen with the pen up. Guide it horizontally across the screen, simulating the sun traveling from east to west. After each incremental movement of the sun turtle, move it to the walker's position, give it an appropriate heading, move it FD a few steps, and return to the sun position.

Perhaps (but not necessarily) the solution could make use of those little known primitives, LABEL and GO.

LCSI Logos can invoke the WINDOW command and set the sun far away (SETPOS [10000 10000]), and then start it moving westward. Sprite Logos, such as Commodore's, make it easier to code a solution; one sprite could be the SUN, while a second becomes the WALKER.

Is your guess about the walker's path borne out by the turtle? Remember that the sun doesn't travel east to west in the summer in the Arctic; it makes a complete circle about the sky. What path would the turtle "walk" if the sun orbited around HOME in a screensized circle?

When you have a working solution (or even a valiant attempt), send your solution, along with a self-addressed stamped envelope to:

NLXual Challenges
Attn: Robs Muir
1688 Denver Avenue
Claremont, CA 91711

We will compile the contributed solutions, select the most interesting or unusual, and redistribute them to each contributor. Since our "challenge community" involves problem solvers from six countries, the contributions are quite diverse. Solutions are available for each problem, but you must contribute your attempted solution (however meager) to receive these "answers."

Robs Muir is a teacher in the Claremont CA Unified School District and Claremont Graduate School. He presents Logo workshops in the southern California area.

The National Logo Exchange, copyright (c) 1984 by Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly, September through May, $25 (US) per year, mailed first class from Charlottesville, VA. $5 (US) additional per year for addresses outside the USA, Canada, and Mexico. The opinions expressed by the authors are not necessarily those of The National Logo Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress Street, Salem, MA 01971. Address other correspondence to: The National Logo Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor..Tom Lough

THE NATIONAL LOGO EXCHANGE

Volume 3 Number 5 FORWARD 100! January 1985

Circles, Arcs, and Headaches

by

Sharon Burrowes

Although we have been using Apple (LCSI) Logo in our elementary schools for more than a year, suddenly we became aware of a problem of the kind that gives a computer coordinator such as myself grey hair, ulcers, and headaches all at once!

The difficulty centered around the built-in arc and circle procedures provided on the Apple Logo master disk. (For NLX readers not familiar with this, there are circle and arc procedures in the STARTUP file on the master disk which can be loaded in after you boot Apple Logo.) Initially, these procedures seemed like a neat idea to speed up programming and make things easier for the students.

HEADACHE # 1

However, the problems started when quite a few students began using these procedures in their work. The circle and arc procedures are buried; that is, they take up space in memory, but are not listed when you type POTS. Thus, students (and many teachers) are unaware that they are not Logo primitives.

My headaches began with a rash of telephone calls about "sick" computers that were actually in excellent health. The problem was not the equipment at all, but rather these circle and arc procedures.

Here's what typically happens: Student 1 boots Apple Logo, takes out the master disk, puts in a file disk, and presses RETURN. (This process does not load the circle and arc procedures into the workspace.) Later, Student 2 comes to the computer, loads a set of procedures which need the circle or arc procedures, and tries to run them. Messages such as I DON'T KNOW HOW TO CIRCLER appear, and telephones such as mine begin to ring!

HEADACHE REMEDY # 1

About the fourth or fifth time this happened, I decided that something had to be done. "Aha! I'll just replace the STARTUP file on the master disk with an empty file, and let the teachers know where the procedures can be found in the manual," I thought to myself, and did so.

My solution brought cries of agony from my more sophisticated teachers who had carefully taught their students how to manage the procedures correctly. They wanted them back and couldn't understand why everyone couldn't teach their students how to load them properly.

HEADACHE # 2

My bright idea of letting students and teachers type in the procedures from the manual brought more cries of agony. Oops! I practically live in front of a keyboard, and had forgotten about the typing speed of fourth graders and teachers with little keyboard experience. My headache was getting worse, and my hair greyer!

I firmly believe that circle drawing is a very important problem solving experience, and that students should explore how to do it for themselves. However, the time had come for a compromise!

HEADACHE REMEDY # 2

I took apart the STARTUP file and isolated the circle and arc procedures. Next, I saved each procedure in a separate disk file named after the procedure. For example, the CIRCLER procedure was saved under the file name CIRCLER. This made it possible for students to load into their workspace the particular circle and arc procedures they might need. This reinforced the idea that a procedure and not a primitive was doing the circular drawing for them.

Headaches continued

Then I designed a series of activity sheets to encourage students to explore on their own how to draw curves and circles. Included were suggestions using REPEAT to explore curved lines, closed figure projects to help students deduce the Rule of 360, and some hints on controlling the size of circles. (Ed. Note: For additional arc and circle activity suggestions, see Tipps for Teachers elsewhere in this issue.)

MORE HEADACHES TO COME?

If you are just getting started with Apple Logo, perhaps you can avoid a few more grey hairs, incipient ulcers, and LOTS of headaches by avoiding my mistakes. Make an appropriate decision about these "built in" circle and arc procedures, and then train both fellow teachers and students in a way which best suits your curriculum. I believe that this particular source of headaches is under control for now, but I wonder what will be waiting for me when I ARCR around the corner!

Sharon Burrowes is the computer coordinator for the Wooster (OH) City Schools, and is a frequent contributor to the NLX and other national educational computing periodicals.

Logo Depository Established

Educators at Saint Martin's College are establishing a depository of Logo materials as a resource for continued support of Logo teachers. They are soliciting contributions of Logo related materials (software, research articles, workbooks, texts, periodicals, "hands on" guides, etc.) to become part of their permanent collection. The resource material will be incorporated into a Logo Library designed to allow teachers to explore Logo software, literature, and curricula. This Logo Library is the object of a current masters project, and will interface with the Saint Martin's College Microcomputer Resource Center, which boasts the largest software collection on the west coast. For more information, contact Diane McMurdie, Curriculum Library, Saint Martin's College, Lacey, WA 98503, or call (206) 491-4700.

From the Editor

Be It Resolved

Each January brings with it an invitation for new year's resolutions. As I was formulating some of mine, I got to thinking about another definition of resolution I learned some time ago. It went something like this:

Resolution - the process of distinguishing between or rendering visible the separate parts of an item under observation.

Granite gives us a straightforward application of this aspect of resolution. When you look at a chunk of granite up close, you notice a "salt and pepper" crystalline texture to its rough surface. The various mineral components are readily visible; you can easily distinguish the icy quartz and sparkly mica from the darker hornblende and the pastel feldspar. At a distance of several meters, however, this same piece of granite appears to be more or less uniformly grayish or pinkish, with a smoother texture. The individual mineral crystals appear to be so small that they are indistinguishable. It's a matter of resolution.

In this issue of the NLX, you find several references to circles. But when we draw a circle, we actually draw a regular polygon. What makes the polygon a circle? We have to draw the sides so small we cannot distinguish them. Only then can we see the full beauty of the form we call the circle. It's a matter of resolution.

In Logo teaching, we are acutely aware of how things seem to be going in our classes from day to day. But sometimes we agonize over details and get deeply involved in specific actions. It is only when we view everything in perspective (at arms length, so to speak), that the specifics begin to merge and we sense the overall progress being made. It's a matter of resolution.

There are times when we want to see the various mineral crystals in granite, the nature of the sides of a polygon, the exact organization of a Logo procedure. But let us resolve not to forget the importance of another point of view, one in which the details fade, allowing more universal characteristics to emerge.

FD 100!

Tipps for Teachers

by Steve Tipps

Wonderful Wiggles

Student projects often require the use of circles and curves. Even after students are able to make a circle using the ubiquitous

```
REPEAT 360 [FD 1 RT 1]
```

they are often perplexed when they need a larger circle, smaller circle, or part of one. Ocean waves, flying birds, species of flowers, and many other projects depend on facility with wiggles. Helping students understand how circles are made is a continuing process in the Logo classroom. Try some of these ideas to see what kind of wonderful wiggles can be made.

CIRCLE SENSE

Circles are the source of many puzzles and projects with Logo. Just making a circle is quite a thinking task. (Ed. note: For a different point of view, see the NLXionary in this month's issue.) For both elementary students and adults, playing turtle shows what the screen turtle does when making a circle. One method is to walk a circle drawn on the floor with tape or chalk. Another is to have students work with Cuisenaire rods or kindergarten blocks to approximate a circle.

Different sizes of Cuisenaire rods are different colors. Make circles with white or red rods (which are short) and with the longer black or orange rods. See how many of each sized rod is required to make an "adequate" circle. Typically, the answer is "at least" ten. Fewer than ten makes a figure which is more accurately labeled a square, hexagon, octagon, or other regular polygon.

The size of the circle can also be calculated by relating it to the equivalent number of white rods. A circle shape formed with twelve red rods is equivalent in circumference to twenty-four white rods. Working with rods reveals that the more sides the figure has, the closer it comes to looking like a circle.

Another understanding is more difficult to model. Turn a little or turn a lot provides intuitive understanding, but it is not easy to quantify. Some students get the Rule of 360 quickly, while others need more structure to discern it. Some partial circle probes can be helpful. Have the students guess what each of the following REPEAT statements draw.

```
REPEAT 10 [FD 5 RT 18]
REPEAT 9 [FD 10 LT 10]
REPEAT 45 [FD 1 RT 3]
REPEAT 24 [FD 4 LT 15]
REPEAT 12 [FD 3 RT 10]
```

Have students test their guesses by typing in statements to see what happens. Many of the guesses will be far off. Have each student (or group) construct a similar REPEAT statement and explore it completely. Make half circles, whole circles, large circles, small circles, quarter circles. Change LEFT to RIGHT and change FORWARD to BACK. Have students keep track of the number of REPEATs, the number of steps FORWARD (or BACK) and the number for the turn commands for the various figures. Then you are ready to probe for a rule about what must be true for angles in circles.

Collect all the information on a chart. Each group of students should have some information to contribute. Here is a sample chart for quarter circles. Use similar charts for half circles and whole circles.

QUARTER CIRCLE

	REPEAT #	FORWARD #	RIGHT #
Group I			
Group II			
Group III			
Group IV			

From the chart, the total turn number (REPEAT # times RIGHT #) needed for a quarter circle, half circle, and whole circle should be apparent for most students. However, actually using the Rule of 360 may not be nearly as easy. Inducing the rule is simpler than the reverse process of deducing what should be done with the rule.

Piaget describes this last step as part of the transition from concrete to formal or hypothetical-deductive reasoning. Inducing the rule is descriptive of what actually happens. Deducing from the rule requires hypothesizing the relationship between variables. One potent line of Piagetian research in Logo would be to see whether students with Logo experience actually make this

Tipps continued

transition earlier, as Papert suggests. (See Mindstorms, page 174 ff.)

Working with REPEAT statements to make partial circles should generate many projects. Making parts of circles is a good exercise in modularity. Which of the REPEAT efforts is the most effective in making good circles?

The circle needs enough sides to appear to be circular in shape. But the number of sides should not make the circle a trial of patience. Creating a whole family of circle parts means the REPEAT number should be easily divisible be 2, 3, 4, 5, or 6. Therefore, 12, 15, 18, 20, 24, 30, or even 36 or 48 REPEATs could be good choices. Here are examples of part circles using 24 in the REPEAT statement.

```
TO CIRCLE
REPEAT 24 [FD 3 RT 15]
END

TO HALF.CIRCLE
REPEAT 12 [FD 3 RT 15]
END

TO Q.CIRCLE.RT
REPEAT 6 [FD 3 RT 15]
END

TO Q.CIRCLE.LT
REPEAT 6 [FD 3 LT 15]
END
```

Notice that the size for all of these partial circles is based on FD 3. The procedures are limited because they are fixed in size. Instead of writing new procedures for every new size, variable procedures for the circle parts are more flexible.

```
TO Q.CIRCLE.RT :SIZE
REPEAT 6 [FD :SIZE RT 15]
END
```

The explorations are 100 times easier and more fruitful. The results can make curlicues, wiggles, shapes, and designs of many kinds.

One additional step in making wiggles seems necessary. Half circles and quarter circles were fine, but what about 3/10 or 0.75 of a circle? New procedures for each new fraction are possible, but a new variable procedure with an input for the part is better. CURVE.RT seems just right because it uses the part of the circle as input.

```
TO CURVE.RT :PART :SIZE
REPEAT :PART * 24 [FD :SIZE RT 15]
END
```

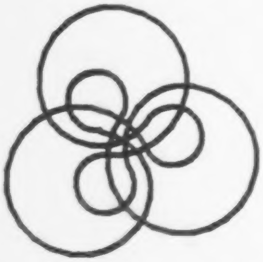
You can write CURVE.LT to draw curves to the left. The variable PART is useful for common fraction input, such as 1/2 and 1/3.

```
CURVE.RT  5/6  4
CURVE.LT  5/6  10
CURVE.LT  1/3  8
```

The above sequence of curves makes a nice wiggle which can be combined into a REPEAT statement

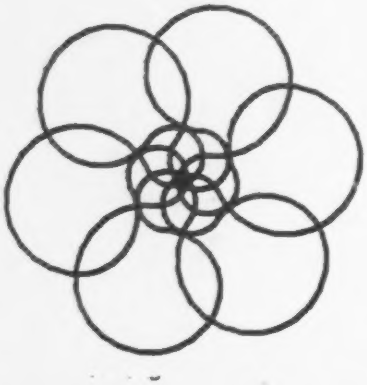
```
REPEAT  3 [CURVE.RT 5/6 4 CURVE.LT  5/6
10 CURVE.LT 1/3 8]
```

to define a TRIFOIL procedure.



A small change in the PART input to the second curve changes the shape into a POSY.

```
TO POSY
REPEAT  6 [CURVE.RT 5/6 4 CURVE.LT  4/6
10 CURVE.LT 1/3 8]
END
```

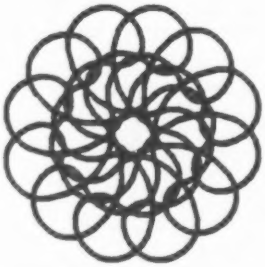


Tipps continued

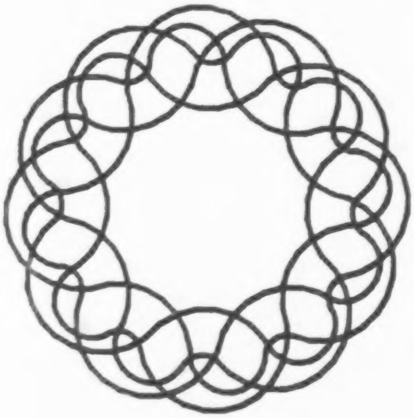
Other combinations and changes also make interesting wiggles and flower shapes. Some figures may close, while others undulate across the screen. It is both fun and instructional to try to predict the outcome of the figure in advance.

Decimal fractions can also be used as the PART value. STAR.FLOWER and LACE.EDGE demonstrate both the use of decimal fractions and how slight differences in the commands make major differences in results.

```
     TO STAR.FLOWER
     REPEAT 12 [CURVE.RT 0.3 3 CURVE.LT  0.5
5 CURVE.RT 0.3 10 CURVE.RT 0.5 5]
     END
```



```
     TO LACE.EDGE
     REPEAT  12 [CURVE.RT 0.3 3 CURVE.LT 0.5
5 CURVE.RT 0.3 10 CURVE.LT 0.5 5]
     END
```



These wonderful wiggles should lead to many more. They are the result of playing with fractional ideas. Whether your students are working with common fractions or decimals, parts of circles are a good way of exploring.

For additional ideas and projects, I suggest The Rule of 360, available for $7.95 from Martin-Bearden, Inc., 1810 Erwin Place, McKinney, TX 75069.

[Note: The procedures in this column were developed with Terrapin Logo and work equally well with the Commodore and Apple MIT Logo versions. However, Apple (LCSI) Logo treats the common fraction input to CURVE.RT and CURVE.LT somewhat differently. Only seven repetitions result from CURVE.RT 1/3 8, for example. But PRINT 1/3 * 24 gives 8, and CURVE.RT 0.33333 8 works correctly. The cause of this inconsistency is not readily apparent. It seems to stem from the method REPEAT rounds or truncates its input to form an integer. Any ideas?]

POLY WANTS AN HOUR (OR TWO)

With complete understanding of the Rule of 360 comes the realization that all polygons can be made from a single variable procedure. By setting the numbers of sides with an input variable, all regular polygons can be drawn.

```
     TO POLY :SIDES :SIZE
     REPEAT :SIDES [FD :SIZE RT 360/:SIDES]
     END
```

While this procedure for POLY is often presented as a formula, Logo teachers must resist the easy way. Many students from ages 10 to 15 can create an all purpose regular polygon maker for themselves. After discovering POLY, students typically go back to specific procedures for SQUARE, OCTAGON, and HEXAGON. This doesn't mean that they have forgotten the lesson of POLY. Instead, it indicates the powerful idea of "naming," which is built into Logo. Many different animals are dogs, but they are called collies, spaniels, and schnauzers, or even Lassie, Irma, and Spritz.

Time is needed for this learning. Avoid being impatient with children and adults who need time to play with Poly. Logo is not a timed test. The digression properly built into Logo activities is one of its potent features because trial and mistake are on the route to understanding.

Steve Tipps is the West Professor of Education at Midwestern State University in Wichita Falls, TX. He presents Logo workshops for school systems throughout the United States.

Krell Logo for the Apple is available for $73.95 from Software Solutions, 11185 San Pablo Avenue, El Cerrito, CA 94530. (415) 237-5257. Include $3 for shipping and handling.

NLXionary

A Lectionary of Selected Logo Readings

by Griff Wigley

"Computing for Families," by Fred D'Ignazio, Compute!'s PC & PCjr, October, 1984.

>Here is an article which illustrates the profound misconceptions of Logo still prevalent in today's world of educational computing.

>D'Ignazio reports that experts in the field of educational computing are increasingly negative on both teaching students about the computer as well as teaching them to program, as means of achieving computer literacy. He says teachers and parents see Logo as a "computer-literacy language for themselves and their youngsters." Since today's computers and programming languages will be obsolete by the 21st century, we may be "miseducating" today's students by teaching them programming and how computers work.

>Then, as sort of an aside on the appropriateness of Logo for children and its influence on children's development and learning, he says, "In a study made at the Bank Street College of Education, for example, researchers found that students who learned Logo did no better academically than those who didn't learn Logo at all." There it is, proof positive. Hold on; there's more.

>D'Ignazio contends that all programming languages, including Logo, are inappropriate for children because so much effort must be extended to achieve even a small effect, "such as directing a little screen cursor (called a turtle) to draw a circle. Children don't perceive this as especially meaningful, since they could do the same thing in a couple of seconds on a piece of paper." And his proof? "(I have two small children and know this to be true from personal experience.)"

>Logo is not practical either, contends D'Ignazio, because, even though it can be used for many applications, it requires too much work. "This is because Logo is essentially a graphics language - a language for drawing pictures on the computer's display screen. Making Logo do more than draw pictures is an inappropriate use of the language. And for most families, it's not worth the effort."

>Since D'Ignazio views Logo only as a tool for drawing pictures, it is not surprising that he recommends (for those families who, for heaven only knows what reason, may still be interested in Logo) turtle graphics packages such as Delta Drawing and Turtle Power, because they "may be a great deal less expensive than Logo, and they may have features that Logo lacks."

>Most of what D'Ignazio says about Logo I could agree with if he had been speaking about BASIC. But, somewhere along the line, he missed the boat on what Logo is all about. I'll take a Koala Pad any day over Logo for just drawing pictures. Yup, even for circles! But what about for understanding circles? Coming to know the mathematical concepts underlying the formation of a circle is a bit tough if all you have is a pencil and paper - or a Koala Pad. These are concepts which have been tossed around since the ancients, and will still be worth learning in the 21st century, using Logo or some other appropriate tool to facilitate the learning process.

>D'Ignazio fails to see how Logo can foster an approach and an attitude in a learner towards learning a subject like math that can easily outlast the life of the language itself. He needs to learn that Logo is a philosophy of education more than just a computer programming language.

>The rest of his article has merit in that it emphasizes using computers as application tools. But interspersed in his list of recommended types of software for families (word processing, filing cabinet, communications, music, painting, etc.) he recommends the purchase of "learning software that turns their computer into a teaching machine." Now I understand how he missed the Logo boat.

"Logo: Fad or Fixture?" by Dan Watt, Popular Computing, January, 1985.

>Watt's subtitle says it all: "Popularity doesn't always reflect a true understanding of purposes." How true it is. Oh, sure, Logo is as popular as ever. But, as Watt notes in his article, the use of Logo as a "computer literacy unit" abounds. Here are some thoughts which come to my mind after reading this provocative article.

>Recently, I attended MECC '84 (the annual conference of the Minnesota Educational Computing Corporation), where one woman spoke proudly of the mobile computer laboratory that traveled around to the dozen elementary schools in her district. All 3rd graders at a school got four weeks of Logo,

NLXionary continued

one hour per day. After that, the mobile lab went to a different school, and the 3rd graders never got to use Logo again.

>This problem is not unique to schools in Minnesota. In too many other systems, having students actually use their new skills is seemingly not important. Exposure and equal opportunity take precedence. And we wonder why students get bored and view school as a trivial experience.

>It is difficult to get people even to talk about Logo's purposes. At the MECC Logo sharing forum, Tom Boe, a MECC trainer and colleague of Marcia Horn, wondered if we were unwittingly creating some kind of monster with our approach to implementing Logo. I wonder what kind of backlash will occur when students, parents, teachers, administrators, and school boards slowly realize just how little effect all this commotion is having. I believe that, unless you have a long range strategy for changing the very nature of how your teachers approach learning, and have the resources to invest heavily in the training and ongoing followup, don't bother implementing Logo. It does Logo and the children a disservice, and, for all the effort required, it's not much fun. Just ask Fred D'Ignazio.

"Sprite Logo," a review by Don Crabb, InfoWorld, December 3, 1984.

>Put another feather in LCSI's hat for its second rave review from InfoWorld within a month. I rejoice that InfoWorld discovered Sprite Logo has many of the features they extolled 3 weeks earlier in their review of Coleco's SmartLogo. I bemoan the lack of collaboration between the two authors. I'm curious why InfoWorld didn't have the same person review both products.

>Logo enthusiasts will not find much substance in this review. Although it includes explanations about Logo in general, and about the materials included in the package, nowhere are the language features described or compared with those of other Logo versions. It is ironic that a highly complimentary review of what I believe to be a fine product seems to do it a disservice.

Griff Wigley is a co-owner of Family Computing, Inc., a facilitator in the Faribault (MN) Public Schools, and a school board member of Prairie Creek Community School.

TurtleTips

by
Donna Lanyi
and Jane Toth

Winter Fun

Are you ready for some teacher time-saving Logo hints? Here is a collection of ideas we have found to be helpful in our classrooms.

TEACHER HINTS

How many times have you heard, "Teacher, I need your help!" but you have been unable to leave the group you were working with? We could all use an extra hand in the classroom to answer Logo questions. One solution: Student Helpers!

Every classroom has a few students who excel above the others. Utilize their knowledge and skills to be your student Logo helpers. Whenever a problem arises and you can't get to the computer, refer it to a student Logo helper.

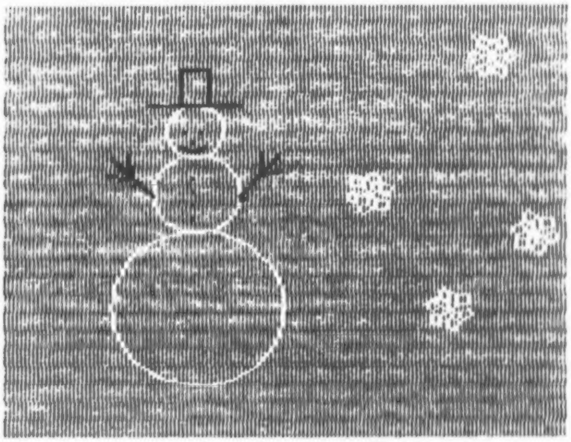
Scheduling the computer in the classroom can at times be more of a hassle than anything else. Some schools are able to utilize a computer lab in their building. Other schools have a computer in classrooms on a rotating basis. A laminated Logo time schedule where the students can sign up or be assigned is a helpful aid. In our personal opinion, the approach that works best is to have the students at the computer for several shorter periods of time more often during a week rather than for longer but fewer periods. Constant reinforcement is essential to grasping Logo concepts.

When introducing a new concept or primitive, we have found using our small groups (reading and math) extremely useful. It allows for an excellent chance to work closely with students. Plus, you already have the groups formed in your classroom.

PROCEDURE OF THE MONTH

Winter fun! There are many geometric shapes in winterscapes, especially if you live in an area where snow falls. Encourage your students to create graphics of winter scenes. Here is one as a starter to help generate ideas. The scene below was given to us by Deborah Bartter, a tutor in Wooster City Schools, here in Wooster, Ohio.

TurtleTips continued



If you are interested in a listing of this procedure, please send us a self addressed stamped envelope. We invite any elementary teachers or students to send their procedures or ideas to be shared. Donna Lanyi and Jane Toth, Beall Avenue School, 716 Beall Avenue, Wooster, OH 44691.

Donna Lanyi and Jane Toth use Logo in their elementary classrooms in the Wooster (OH) City Schools.

Logo Disserts:

Dissertations Dealing with Logo

by Barbara Elias

To what extent does a child's developmental level influence the ability to write procedures using Logo? In an exploratory study carried out in 1972, M. J. Folk investigated the relationship between developmental characteristics of children and how they might affect the ability to learn Logo programming concepts.

The developmental stages as described by Piaget served as the theoretical framework for the study. As children progress thought the identified states, they are able to deal with the higher level thought processes. Thus, the formal operational thinker would be more likely to be successful with recursion concepts in programming while the child who is at the level of concrete operational thought would be expected to be successful with less complicated programming tasks.

From a set of rather general relationships, as expressed in the recursion example, Folk developed a set of twenty-eight specific relationships or conjectures. Each conjecture expressed an expected linkage or similarity between performance on Piagetian tasks and performance on Logo programming tasks. He then investigated the validity of the proposed linkages in a five month study involving eight 4th grade children.

Programming tasks were grouped around five categories:

1. Names and things
2. Procedures
3. The concept of "state"
4. Conditionals
5. Recursion

The children attended three 45-minute sessions each week. They were instructed in the language and provided projects which utilized the programming concepts. Performance on developmental tasks was compared to performance on Logo tasks. The results indicate that the developmental level is a factor in learning to use some of the programming concepts.

In fact, the developmental level was a better predictor of performance in the areas included in this experiment than IQ tests and achievement tests.

From a more practical perspective, the results of this study indicate that the concepts of variables, conditionals, and recursion caused the most difficulty for the 4th grade children. They were, however, able to make programs work in all of these areas, but their understanding did not extend to formal type usage of these concepts.

>>>Folk, M.J. (1973). Influences of Developmental Level on a Child's Ability to Learn Concepts of Computer Programming. (Doctoral dissertation, Syracuse University, 1972). Dissertation Abstracts International, 34, 1125-A.

(Ed. note: Each month, Barbara Elias highlights a dissertation or thesis dealing with Logo. She is accepting copies of recent research results for the NLX Dissertation and Thesis Repository. Reports may be mailed to her in care of NLX, PO Box 5341, Charlottesville, VA 22905.)

Barbara Elias is an assistant professor in the Education Department of Virginia State University in Petersburg, VA, and a doctoral candidate at the University of Virginia.

Listful Thinking

by
Glen Bull and Paula Cochran

Adding Delete To Your Tool Kit

By the time you get this month's NLX, you'll probably be wanting to delete some things from your list of New Year's resolutions. We'll show you how to do that, BUTFIRST, let's look at a game called Sentence Builder.

SENTENCE BUILDER

To play the game, first make a phrase.

```
?MAKE "PHRASE [THE SPARKLING SNOW]
```

Now ADD a word to the phrase. (ADD is a list tool from the December column. The procedure listing appears at the end of this column.)

```
?ADD "FELL "PHRASE
?PRINT :PHRASE
THE SPARKLING SNOW FELL
```

The object of the game is to keep adding words that will eventually make a sentence.

WORD BUILDER

A tougher game is to keep adding letters to a word. The ADD procedure at the end of the column this month is improved so that you can add letters to a word, as well as words to a list. If you typed in ADD last month, you will want to edit it to include this new capability.

We'll illustrate the game with a sample session between Tim and Sue. Tim begins by making a name with one letter.

```
?MAKE "NAME "T
```

Then Sue has to ADD a letter.

```
?ADD "A "NAME
?PRINT :NAME
TA
```

If Sue is challenged, she has to tell the word she is thinking of. In this case, she might have been thinking of "take". The object of the game is to avoid being the person who finishes the word.

To make the game easier, we could write a procedure called ATTACH. This procedure adds a letter and prints the new result, saving you and your students some typing.

```
TO ATTACH :LETTER
ADD :LETTER "NAME
PRINT :NAME
END
```

Now Tim can add a letter this way.

```
?ATTACH "I
 TAI
```

Tim could have been thinking of "tail". However, Sue can't add the letter "l", because that completes the word. Instead she attaches "n".

```
?ATTACH "N
 TAIN
```

If Sue is challenged, she has to tell what word she is thinking of. If Sue doesn't have a word in mind, she loses. If she is thinking of a real word, the challenger loses. In this case, Sue could have been thinking of "taint". Now it's Tim's turn. Can you think of a letter Tim could add?

This word game can be a great spelling and vocabulary builder. It is also good for inspiring dictionary use. (The dictionary should be consulted only when a word is challenged.)

IMPORTANT WORDS

Not all words are created equal. Some words in a sentence are more important than others. English teachers tell us that nouns and verbs are important ones. But how can they get across the idea of grammatical categories? Diagramming sentences doesn't always work!

Here's a way using Logo. It uses a procedure called DELETE. DELETE is another list tool, just like ADD, PICK, and INSPECT. First make a sentence.

```
?MAKE "PHRASE [THE BIG BROWN BEAR
  CAUGHT A TROUT]
```

283

Listful Thinking continued

Now use the DELETE procedure to delete a word. (You will find the actual DELETE procedure at the end of the column.) Here is how to DELETE the word "BROWN from the phrase.

```
?DELETE "BROWN "PHRASE
?PRINT :PHRASE
THE BIG BEAR CAUGHT A TROUT
```

After "BROWN is removed, the remaining words still make a sentence. The object of the game is to avoid destroying the sentence. Let's delete another word.

```
?DELETE "BIG "PHRASE
?PRINT :PHRASE
THE BEAR CAUGHT A TROUT
```

Almost every word that can be deleted has been removed, but one more can be taken out.

```
?DELETE "A "PHRASE
THE BEAR CAUGHT TROUT
```

A more advanced class will recognize that both "a" and "the" are articles. In this class, we could raise the question:

> "Why is it ok to take out "a",
> but not "the"?

In other words, THE BEAR CAUGHT TROUT is a working sentence, but BEAR CAUGHT A TROUT is not.

Is this because "a" is different from "the"? Does it have to do with whether the article is at the beginning of the sentence, or perhaps the kind of noun the article precedes? These are all good questions for students acting in the role of linguistic investigators.

Now it's Tim's turn again.

```
?DELETE "TROUT
?PRINT :PHRASE
THE BEAR CAUGHT
```

Uh-oh. Tim loses, because THE BEAR CAUGHT isn't a working sentence. Of course, he could argue that it is a sentence, because it has a subject and a predicate. That would be fine, since it leads to a discussion of what it takes to make a complete sentence, the role of transitive and intransitive verbs, and lots more. That was probably what the teacher had in mind all along!

COPY CAT

One characteristic of DELETE is that it gradually destroys the phrase you are working with. You could type in the list again, but it might be a long one. An alternative is to make a copy of the list and work with that, if you don't want to permanently destroy it.

```
TO COPY :LIST
MAKE "COPY :LIST
END
```

Assume that we have this phrase:

```
?PRINT :PHRASE
THE BIG BROWN BEAR CAUGHT A TROUT
```

We can make a copy of it by using the COPY procedure.

```
?COPY :PHRASE
?PRINT :PHRASE
THE BIG BROWN BEAR CAUGHT A TROUT
?PRINT :COPY
THE BIG BROWN BEAR CAUGHT A TROUT
```

Now when we DELETE a word from the copy, it won't affect the original phrase.

```
?DELETE "BEAR "COPY
?PRINT :COPY
THE BIG BROWN CAUGHT A TROUT
?PRINT :PHRASE
THE BIG BROWN BEAR CAUGHT A TROUT
```

If we don't like the result of the deletion, we can make another copy of the original to work with.

PLAYING THE GAME

There are a couple of things we can do to make it easier to play the game of deleting words from sentences. To start playing, we have to make a sentence.

```
?MAKE "PHRASE [THE FROSTY SNOWMAN IS A
   VERY MERRY SOUL.]
?PRINT :PHRASE
THE FROSTY SNOWMAN IS A VERY MERRY SOUL.
```

That's fine if your students know about MAKE statements. If they don't, an easy way for them to make sentences is needed. Here's a procedure to do that. In Terrapin Logo substitute the command REQUEST in place of READLIST.

Listful Thinking continued

```
TO GAME
PRINT [TELL ME A SENTENCE.]
MAKE "PHRASE READLIST
PRINT [NOW TAKE TURNS REMOVING]
PRINT [WORDS FROM YOUR SENTENCE:]
PRINT :PHRASE
END
```

When we DELETE a word from a list, we have to use PRINT to see the result. (Incidentally, does it matter whether we delete VERY or MERRY first?)

```
?DELETE "VERY "PHRASE
?PRINT :PHRASE
THE FROSTY SNOWMAN IS A MERRY SOUL.
```

Here is an enhancement which deletes the word, and prints the result in one step, saving you some more typing.

```
TO REMOVE :ITEM
DELETE :ITEM "PHRASE
PRINT :PHRASE
END
```

Here's how to use REMOVE.

```
?REMOVE "FROSTY
 THE SNOWMAN IS A MERRY SOUL.
```

We hope that reading this column has REMOVEd all thoughts of your New Year's resolutions, at least for a few minutes. Next month we will give you a list tool that will allow you to SELECT new ones!

Here are Apple Logo and IBM Logo procedures for ADD and DELETE:

```
TO ADD :ITEM :LIST
TEST WORDP THING :LIST
IFFALSE [MAKE :LIST LPUT :ITEM THING  :LIST]
IFTRUE  [MAKE :LIST WORD THING :LIST :ITEM ]
END

TO DELETE :ITEM :LIST
MAKE :LIST DELETE.LOOP :ITEM THING :LIST  []
END

TO DELETE.LOOP :ITEM :OLDLIST :NEWLIST
TEST EMPTYP :OLDLIST
IFTRUE [PRINT SE :ITEM [NOT FOUND]]
IFTRUE [OUTPUT :NEWLIST]
TEST :ITEM = FIRST :OLDLIST
IFTRUE [OP SE :NEWLIST BF :OLDLIST]
OP DELETE.LOOP :ITEM BF :OLDLIST LPUT
      FIRST :OLDLIST :NEWLIST
END
```

Here are Terrapin Logo procedures for ADD and DELETE:

```
TO ADD :ITEM :LIST
TEST WORD? THING :LIST
IFF MAKE :LIST LPUT :ITEM THING :LIST
IFT MAKE :LIST WORD THING :LIST :ITEM
END

TO DELETE :ITEM :LIST
MAKE :LIST DELETE.LOOP :ITEM THING :LIST []
END

TO DELETE.LOOP :ITEM :OLDLIST :NEWLIST
TEST :OLDLIST = []
IFTRUE PRINT SE :ITEM [NOT FOUND]
IFTRUE OUTPUT :NEWLIST
TEST :ITEM = FIRST :OLDLIST
IFTRUE OP SE :NEWLIST BF :OLDLIST
OP DELETE.LOOP :ITEM BF :OLDLIST LPUT
      FIRST :OLDLIST :NEWLIST
END
```

Glen Bull is a professor in the University of Virginia's School of Education, and teaches Logo courses at both the graduate and undergraduate level. Paula Cochran is a reformed English major who studied linguistics at the University of Cambridge. She is now a speech-language pathologist working with language-disabled children.

Logo Notes

The NLX and the University of Virginia have been testing an electronic Logo bulletin board system (BBS) during the past several months. The BBS is configured with sections for each of several types of Logo teaching concerns and issues, and allows participants to communicate with each other anonymously, is desired. While the system is not yet open to the general public, if you are interested in gaining access to the BBS, write for more details. Elaine Bartley, Logo Bulletin Board, Department of Curriculum and Instruction, School of Education, University of Virginia, Charlottesville, VA 22901.

The College of Education of Arizona State University announces its Fifth Annual Microcomputers in Education Conference, to be held in Tempe, AZ, March 13-15, 1985. For more information, write to Gary Bitter, College of Education, Arizona State University, Tempe, AZ 85287.

Teacher to Teacher

Guides for Teaching Logo to Young Children

reviewed by
Anne Cairns Federlein

EZ Logo, Minnesota Educational Computing Corporation (MECC), #975, 3490 Lexington Avenue North, St. Paul, MN 55112. Includes a diskette of EZ Logo and a 79 page notebook of suggested activities, work sheets, and lesson plans.

AUDIENCE: Young children, ages four to eight.

PURPOSE: Allows young children to direct their own learning of Logo.

PREPARATION LEVEL OF TEACHERS: The teachers can learn along with the children.

FORWARD WITH HUMAN TURTLES

STRENGTHS: EZ Logo prepares the child to use the Logo language at a later time by introducing limited one stroke commands for the turtle. Initially, it is suggested that the teacher introduce the child to EZ Logo through concrete experiences, such as giving the RIGHT, LEFT, FORWARD, BACK, and CLEAR commands to a "human" turtle. When this activity is understood, then this information is transferred to the turtle on the computer screen. Important skills developed are trial and error problem solving, alternative thinking strategies, and decision making skills, in addition to reading readiness and pattern recognition skills.

WORK SHEETS — WHY?

CONCERNS: Activity sheets are used to repeat the learning done on the computer and to give a record for each lesson. The authors suggest that these sheets not be treated as work sheets, but rather to incorporate them into a journal to record the turtle movements. Children learn by doing — but not a ditto sheet! The mere presence of activity sheets can encourage rote learning and drill and practice teaching methods.

REACTION: Overall, this is one of the best books I have found on the market for teaching Logo to young children. It is easy to follow and easy to learn for both child and teacher. Most importantly, the book introduces early childhood learning theory — that is how to involve young children effectively in their learning. Publishers need creative teachers who understand child development theory to team up with authors of Logo texts to focus on creating such stimulating environments.

Horn, Carin, and Collins, Carroll. Com-Lit Computer Literacy for Kids. Sterling Swift Publishing Company, 7901 South H-35, Austin, TX 78744.

AUDIENCE: Children ages seven through nine.

PREPARATION LEVEL OF TEACHER: Teachers just beginning to learn Logo.

ASK THE RIGHT QUESTION

STRENGTHS: This text is easy to follow by primary aged children. For example, the illustrations throughout the book allow the child to "read" the book and understand the content even if the words are too difficult. Also, the book is spiral bound, and lies flat on the desk, making it easier to use alone or in group work.

The format of the book includes philosophical questions, such as, "Does a computer solve problems just like a human solves problems?" The answer is easy to read and understand. "NO ... because a computer does not have any feelings or emotions. People have emotions that help them make choices. A computer can do only what it is told to do!"

HIDE THE ANSWERS

A suggestion to the authors is that the answers to these questions be placed on another page out of view of the child. This will encourage more discussions in the classroom. When children are presented with such answers, they often just recite them, and do not necessarily understand them. There is no one "right" answer, and, as an experienced teacher, I know that many answers children give to open ended questions are thought provoking and often set the stage for problem solving and creative thinking.

Teacher to Teacher continued

NOT MORE WORKBOOKS!

CONCERNS: The appearance of the book shouts WORKBOOK! Such directed activities can be helpful for the beginning Logo teacher in teaching the new subject. However, it takes a creative teacher to risk leaping from the "canned" lessons to those that address the developmental needs of *each* child in the class. For example, at the beginning of the book, the text begins with an introduction to the computer machine by giving a long vocabulary list of new words to be learned before the child ever touches the computer. Who thinks that learning unknown vocabulary words makes Logo easier?

This format, used in many elementary science textbooks, asks children to memorize the new words of each lesson before they experience what they are! We must question what is best for the child and introduce the vocabulary *after* each child has had an opportunity to play with Logo and experience the concepts. The new words would then make sense and be understood, rather than memorized and forgotten after the lesson is over. Young children must be actively involved in doing *before* there is understanding.

YOUNG CHILDREN ARE DIFFERENT

REACTION: If learning is to be meaningful for children and provide a base for future learning, then teachers must be able to allow children to participate actively in classroom learning. Unfortunately, many of the new Logo books and curriculum materials for the early childhood age group directs the child through a series of pencil tasks. Young children learn by doing, that is, through experiencing, exploring, and manipulating. This is not understood by many authors and teachers who are not trained in early childhood. In the past, this structured way of teaching has been for the convenience of the adults in the classroom, rather than for the children.

Sherry Turkle discusses the effects of computers on children and adults in her book, *The Second Self*, in an interview in *The Chronicle of Higher Education* (November 14, 1984). "I want educators to accept and exploit the fact that children can master the computer in many different ways. We have to get the widest variety of children to use this personal learning instrument in a positive developmental way. Too often, teachers are not open enough to children who don't seem to be serious or organized - who appear to be playing around with the computer, when in fact they're really finding their own way into the scientific and technical culture."

"The message I'm trying to get out is that the computer doesn't just do something," she says. "It does something TO US. And we'd better be more interested."

Perhaps teachers, especially those working with younger children, had better be more interested in this new machine and try NEW teaching methods. It is easier and more convenient to apply the old teaching methods of drill and practice to something new, rather than FIRST watching and listening to the children, and THEN designing an effective curriculum.

OH NO! WHERE ARE THEY?

As I search the book stores, computer stores, libraries, and schools, it is difficult to find materials and books for Logo that are designed to encourage the discovery method of learning. I am becoming more fearful that the new and exciting teaching tool called Logo is falling into the sea of curriculum mediocrity of drill and practice!

SPEAK UP, TEACHERS!

We in Logo Information For Teachers (L.I.F.T.) are interested in hearing from teachers at all grade levels who are integrating Logo materials effectively into the curriculum. Please send your comments along with your name and phone number, so that we can interview you and share your ideas in future columns. L.I.F.T., Post Office Box 5396, Plymouth, MI 48170.

Anne Cairns Federlein is a professor in the Early Childhood Area, School of Human and Educational Services, Oakland University, Rochester, MI.

Computer Contest Announced

The University of Wisconsin-Parkside announces its Fifth Annual Computer Problem Solving Contest on April 27, 1985. Student teams compete in one of three divisions: grades 4-6, grades 7-9, and grades 10-12. In the past, there have not been many solutions using Logo. Here is an opportunity to change that. If you are interested in sponsoring a local contest site, write to: Donald T. Piele, ICPSC, Box 2000, University of Wisconsin-Parkside, Kenosha, WI 53141, or call (414) 553-2231.

Q and A

by
Jim McCauley

Q. I'm having trouble using INSTANT with my students because they hold the key down so long. The keyboard auto-repeat feature causes too much to happen. Is there some way to fix INSTANT so that it takes only the first key press and not all the auto-repeated ones?

<[-?-]>

A. This question has caused me to search down any number of alleyways, both in Logo and in the hardware of the Apple IIe. (That must be the computer you are using, I assume.) The answer to the problem is simple, but risky.

The auto-repeat feature of the Apple IIe keyboard is called "n-key rollover," and it was viewed as a desirable feature when the Apple II+ was redesigned. Usually, when Apple (and many other computer manufacturers) include a feature in a new computer model, they make some provision to deactivate the new feature so that the new model can be made to act like the old one. Often, they create something called a "soft switch" (a special location in the computer's memory) which, when accessed, enables or disables the feature.

This is how the Apple controls many of its hardware features. For example, if you want to make the Apple's speaker click:

```
TO CLICK
IGNORE .EXAMINE 49200
END

TO IGNORE :X
END
```

If you want a "raspberry" from your Apple, try:

```
TO RASPBERRY
REPEAT 40 [CLICK]
END
```

The Apple also controls its text and graphics screens with .EXAMINE and .DEPOSIT, Logo's equivalent to BASIC's PEEK and POKE. For a virtuoso example in Logo, take a look at the LOWRES file in the Apple Logo Tool Kit, a public domain set of software from Logo Computer Systems, Inc. It's a set of routines which allows the use of low resolution block graphics from within Logo.

Unfortunately, Apple didn't design a soft switch for the n-key rollover feature. There is no way to correct fully the misbehavior of INSTANT from within Logo. If this is really a serious problem for you and your students, you are going to have to modify the Apple hardware.

If that sounds scary, that is because it <u>is</u> a bit scary. If your Apple IIe is still under warranty or if you have a service contract, you should do nothing mentioned hereafter. On the other hand, if you really and truly want to defeat the auto-repeat feature of the Apple IIe, here is what must be done:

1. Turn off the power, but leave the Apple plugged in.
2. Take off the cover and touch the power supply (the large metal box to the left of the circuit board) with your fingertips. You won't get a shock. This lets any static electricity in your body run to a safe electrical ground, and helps prevent damage to the computer components.
3. Find the large chip labeled 3600-PRO; it is located in the lower right corner of the Apple IIe circuit board. (It may be labeled with a variation of this code, such as AY-5-3600-PRO.) Note that the end of the chip nearest the keyboard has an orientation mark (usually a white dot or a semicircular depression).
4. Using a "chip puller" (a special tool designed for the purpose), carefully remove the 3600-PRO chip. This can be difficult, because the chip has a lot of pins.
5. With the orientation mark facing you, count up the right side of the chip until you come to the fifth pin. Bend that pin out carefully and gently so that when you reinstall the chip, this pin will not go into its socket.
6. Reinsert the 3600-PRO chip into its socket with the orientation mark facing the same direction in step 3.

If this sounds like a job for a specialist, you are right. In fact, I do not recommend that you do this unless you are absolutely sure that the auto-repeat feature presents a serious problem to your students. It may not be possible to reverse the process, since the pin is fragile, and can be broken off quite easily.

Q and A continued

(Ed. Note: If you have a question about Logo programming techniques or the thinking which leads to Logo programming, send it to NLX Q&A, PO Box 5341, Charlottesville, VA 22905. Each month, Jim McCauley answers selected questions in this column.)

Jim McCauley is a Coordinator of Computer Education for the Santa Clara (CA) County Office of Education, and has written Logo articles for many national publications.

Logo Scouts Wanted

The Chicago Logo Users Group is collecting Logo articles for listing in the bibliography for the Logo 85 conference. If you read an educational or computing periodical not listed below, please register as a Logo article "scout" for that periodical.

Here are the periodicals which are already being scouted.

AEDS Journal
AEDS Monitor
Apple User (formerly Windfall)
Arithmetic Teacher
Byte
Call A.P.P.L.E.
Classroom Computer Learning
Creative Computing
CUE Newsletter
Educational Computing
Educational Technology
Electronic Education
Electronic Learning
inCider
InfoWorld
Journal for Research in Mathematics Education
Journal of Computers in Mathematics and Science Teaching
Mathematics Teacher
Microcomputing
PC World
Personal Computing
Popular Computing
School Science and Mathematics
Teaching and Computers
The Computing Teacher
T.H.E. Journal

If you would like to help locate Logo articles in periodicals not listed here, send your name, address, and periodical name to Andy David, 5430 South Harper, Chicago, IL 60615. Thank you!

This One's for Cary

A workshop entitled "Teaching Physics with the Logo Computer Language" will be presented by Tom Lough and James Hurley in Toronto, Ontario, on January 19, 1985, in conjunction with the joint winter meeting of the American Association of Physics Teachers (AAPT) and the American Physical Society (APS). For more information, write AAPT, 5110 Roanoke Place, Suite 101, College Park, MD 20740, or call (301) 345-4200.

Closing The Gap announces the publication of "Computer Technology for the Handicapped," the selected proceedings from the outstanding 1984 CTG conference held in September 1984. Consisting of 260 pages, the book promises to be a treasury of information for special education and rehabilitation professionals. To order, send $17.95 to CLosing the Gap, PO Box 68, Henderson, MN 56044.

NLXual Challenges

by
Robs Muir

Notes

It never fails.

You have been asked to give a presentation of Logo to a friend/staff/district. Of course, you want to include the "Logo experience" that your students have every day in your class, so you provide some appropriate challenges for your colleagues. At the end of a great six hours of chalk-talk, philosophizing, and laying-on-of-hands (a.k.a. hands-on), you begin to see the spark of enthusiasm on everyone's faces as they rapidly begin to tame the turtle. Everyone except the person in tbe back of the room. As you close, asking for any final questions, you see this person slowly rise and ask what you hoped never to hear, "So what?"

WHAT'S THE BEEF?

Hidden behind this seemingly innocuous eight-character string is a question of awesome import. Consider, if you will, some of the issues raised by this question.

NLXual Challenges continued

1. How does Logo fit into the (capital C) Curriculum?
2. Why should I change my 10 year old lesson plans to include Logo computer programming?
3. Problem solving? My students won't even read the textbook!
4. Games and fun?! Not in my classroom! Schooling is serious business!
5. Ad etceteram.

Since we are all grown up (aren't we?), some of us have a very difficult time playing games, wasting time, and being childish. If you feel guilty when you work on some of the NLXual Challenges (I really should be grading those papers...), perhaps you need a problem worthy of your time ... a problem whose solution could affect a student's entire life ... a really serious problem.

A MOST WORTHY CHALLENGE

Cory, age 9, wants to control his turtle. He knows about Instant Logo programs; other students in his class draw things using the F, R, B, and L keys. However, Cory is unable to move from his neck down due to birth defects. He can move his head just enough to operate a single leaf-switch which, in turn, activates the paddle button on his school's computer.

PROBLEM: How would you write a Logo program to give Cory a turtle experience?

MIT and LCSI versions of Logo support a primitive called PADDLEBUTTON or BUTTONP, which outputs either "TRUE or "FALSE if a specified paddle button is pressed. An example of the use of this command might look like this:

```
IF (PADDLEBUTTON 1) THEN FD 10
or
IF BUTTONP 1 [FD 10]
```

These commands work for Apple joystick buttons, paddle buttons, or Koala Pad buttons. If you use a Commodore 64, you might choose to use the JOYBUTTON primitive for your joystick.

If you don't have the luxury of owning one of these peripherals, you can still write your program using the RC? or KEYP primitive. These primitives output "TRUE or "FALSE if any keyboard key has been pressed — effectively simulating a button press.

What is a Logo experience? Can you have it only with a computer and a turtle, or only if you can type? What kind of Logo experience can you design for someone who can only press one particular key?

Help give Cory, and others, a more playful New Year!

When you have a working solution (or even a valiant attempt), send a listing of your work and results, along with a stamped self addressed envelope to:

NLXual Challenges
Attn: Robs Muir
1688 Denver Avenue
Claremont, CA 91711

We will compile the contributed solutions, select the most interesting or unusual, and redistribute them to each contributor. Since our "community" already involves NLX subscribers in six different countries, the contributions are quite diverse. Solutions are available for each problem, but you must contribute your attempted solution (however meager) to receive these "answers".

Robs Muir is a teacher in the Claremont CA Unified School District and Claremont Graduate School. He presents Logo workshops in the southern California area.

The National Logo Exchange, copyright (c) 1985 by Posy Publications, a part of The Posy Collection, all rights reserved.

Published monthly, September through May, $25 (US) per year, mailed first class from Charlottesville, VA. $5 (US) additional per year for addresses outside the USA, Canada, and Mexico. The opinions expressed by the authors are not necessarily those of The National Logo Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress Street, Salem, MA 01971. Address other correspondence to: The National Logo Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor.......................................Tom Lough

THE NATIONAL LOGO EXCHANGE

Volume 3 Number 6 FORWARD 100! February 1985

Wordles and Turtles

by

Cathy Frank

"Mom, look at my wordles!" My fifth grade daughter proudly held up for my inspection a page full of work. At first, I didn't understand just what a "wordle" was supposed to be. But once I caught on, I became as enthusiastic as she about them.

You see, I enjoy introducing students to the "other half" of Logo, word and list operations, well before they have mastered all the turtle commands. Among other things, this enables me to use Logo with several curriculum areas besides the obvious ones.

However, it is not always easy to think of effective introductory projects in list processing. Some are too simple and unimaginative. Others are too difficult, requiring knowledge of several sophisticated commands. After my daughter's enthusiastic explanation, I realized how much fun, open ended, and challenging "wordles" would be as Logo list processing projects!

OK, WHAT'S A WORDLE?

A wordle is an incomplete word or group of words whose full meaning is implied by the way the words or letters are written. A simple example is

T
O
W
N

The intended word is DOWNTOWN, DOWN being implied by the way the letters of TOWN are printed down the page. Another more challenging example is

RIGHT DINNER

which is wordle-eze for RIGHT BEFORE DINNER! The possibilities are endless, as are the opportunities to use a variety of Logo list processing commands in procedures to display wordles.

The two examples given above use only the PRINT command.

```
TO WORDLE.1          TO WORDLE.2
PRINT [T]            PRINT [RIGHT DINNER]
PRINT [O]            END
PRINT [W]
PRINT [N]
END
```

It didn't take my students long to write another possibility for DOWNTOWN which used the REPEAT statement they had learned with the turtle.

```
TO WORDLE.3
REPEAT 4 [PRINT [TOWN]]
END

TOWN
TOWN
TOWN
TOWN
```

Another simple wordle is

```
TO WORDLE.4
PRINT [STAND]
PRINT [I DON'T]
END
```

I'll let you figure that one out.

If you use Terrapin Logo, PRINT [' '] is useful.

```
TO WORDLE.5
PRINT ['      T      ']
PRINT ['    *   H    ']
PRINT ['  *       E  ']
PRINT ['*           *']
PRINT ['  K       C  ']
PRINT ['    C   L    ']
PRINT ['      O      ']
END
```

Wordles continued

The CURSOR (or SETCURSOR) command helps expand wordle possibilities. Here are a couple of examples.

```
TO WORDLE.6
CURSOR 10 10 PRINT [S]
CURSOR 12 11 PRINT [T]
CURSOR 14 12 PRINT [A]
CURSOR 16 13 PRINT [I]
CURSOR 18 14 PRINT [R]
CURSOR 20 15 PRINT [S]
END
```

```
S
  T
    A
      I
        R
          S
```

```
TO WORDLE.7
CURSOR 14 10 PRINT [P]
CURSOR 14 11 PRINT [U]
CURSOR 14 12 PRINT [Z]
CURSOR 11 13 PRINT [PUZZLE]
CURSOR 14 14 PRINT [L]
CURSOR 14 15 PRINT [E]
END
```

```
   P
   U
   Z
PUZZLE
   L
   E
```

Hardcopies of the above results can be printed, of course, by typing OUTDEV 1 (or (or .PRINTER 0) when you are finished.

DYNAWORDLES

As my students and I developed more examples, it occurred to me that there are wordles one could program which are unique to Logo and the computer; they would not be effective on paper. Here is an example which incorporates subprocedures, the REPEAT and CURSOR commands, variables, and recursion.

```
TO WORDLE.8
REPEAT 30 [FLASH]
END

TO FLASH
CURSOR 20 10 PRINT [AND ON]
WAIT 30
CURSOR 20 10 PRINT [ ]
WAIT 30
END

TO WAIT :TIME
IF :TIME = 0 STOP
WAIT :TIME - 1
END
```

What makes this exercise exciting to me is that writing wordles is fun and challenging with or without Logo. But doing it with Logo expands the wordle possibilities and adds a motivation for learning some of the word and list processing commands whose value is not always appreciated by the beginning list processor. Yet, the challenge is unlimited and open ended and can lead into the use of some rather sophisticated Logo commands.

To wordle a popular expression of a certain newsletter editor,

```
TO WORDLE.NLX :A
IF :A = 35 STOP
CURSOR :A 10 PRINT [100!]
WAIT 5
CURSOR :A 10 PRINT [ ]
WORDLE.NLX :A + 1
END
```

Just in case you had some trouble deciphering the wordles: WORDLE.4 - I don't understand, WORDLE.5 - around the clock, WORDLE.6 - downstairs, WORDLE.7 - crossword puzzle, WORDLE.8 - off and on, WORDLE.NLX - FORWARD 100!

(Ed. Note: For additional related ideas, see also "Is There Logo After Turtle Graphics?" Classroom Computer News, April 1983, and "Tipps for Teachers" and "Microworlds" in the February 1983 issue of The National Logo Exchange.)

Cathy Frank is a computer education consultant in Burlington, VT, and teaches Logo to 5th and 6th graders.

Pascal Newsletter Available

If you are using Logo as an introduction to Pascal, you may be interested in the Advanced Placement Computer Science (APCS) Newsletter published by Tim Corica. Its ten issues per year address Pascal questions, the AP exam, utilities, and the like. For more information, write Tim Corica, APCS Newsletter, The Peddie School, Hightstown, NJ 08520.

From the Editor

Stirring the Pot

One day recently, I stomped into the house half frozen. Posy had just made a pot of vegetable soup and she pressed a bowl of the steamy stuff into my grateful hands as I sank into a recliner. Ah, bliss. When I am subjected to such a dreamlike state, my mind often takes on a life of its own.

I began thinking about soup.

The soup I was enjoying had many different ingredients: tomatoes, potatoes, peas, beans, corn, chunks of meat, seasoning. Although I was aware of each different ingredient as I savored it, there was was a single taste sensation for the soup itself.

How did this come about? While Posy was cooking the soup, she had to stir it often so that it was heated evenly and nothing stuck to the pot and burned. Some of the flavor of each ingredient was absorbed by the others, but individual identities were retained.

But what does this have to do with Logo?

The Logo community is certainly made up of many different types of individuals. They share a common interest in the future of educational computing with this powerful language.

But, unless the "Logo broth" is stirred frequently, it may not pass the taste test. I'd like to encourage the healthy exchange of differing points of views, the airing of opinions and recommendations. For example, John Green's article, "Logo Under Fire," in the January issue of Classroom Computer Learning, gives an excellent focus. It contains an interview with Seymour Papert and suggests that confronting controversial issues is crucial.

But more such articles is not the full answer. Action is needed. Action such as more direct communication between Logo folk, at conferences and workshops, in faculty workrooms (I refuse to call them "lounges") and classes. Action such as letters to authors, publishers, manufacturers, with detailed suggestions, comments, complaints. Action such as each of us assertively trying to find the best way for us and our students to take advantage of the potential of Logo.

Stir the pot a little; it makes a better soup.

FD 100!

Tipps for Teachers

by Steve Tipps

Survey of Surveys

Every newspaper, magazine, and news broadcast seems to include a survey of some kind. What are the best sellers? Who is winning the election? How many people have quit using salt? Everybody likes to have their opinion counted, including students. This self interest can spark many classroom activities using surveys.

Surveys have two main parts: data collection and data display. Logo can be used for both.

GATHER YE INFORMATION

Many elementary mathematics texts include a section on surveys. Students ask their classmates about things such as favorite pets, TV shows, records, colors, etc. Classroom surveys can also be taken on kinds of shoes, colors of eyes, types of transportation to school, months of birth, and 1001 other topics of personal interest. Although the approach may differ, the topics can be used in kindergarten, sixth grade, high school, or college.

In taking a survey, students scurry around the room asking classmates questions. The first attempt may be a quite a mess, but, after this experience, the need for precise data recording is obvious.

Additionally, the way the question is asked makes a lot of difference in the results. The difference between open and fixed categories is clear. An open question is, "What is your favorite color?" The fixed or closed form asks, "Is your favorite color red, blue, green, or yellow?"

If the students want to ask the question of many people, an interactive computer program might be a good tool. Here is a probe for the favorite breed of dog.

```
WHICH IS YOUR FAVORITE KIND OF DOG?
     POODLE
     SPANIEL
     SHEPHERD
     COLLIE
     POMERANIAN
     OTHER
TYPE IN YOUR ANSWER.
```

Tipps continued

Here is another for birthday month.

```
IN WHICH MONTH WERE YOU BORN?
PLEASE TYPE IN YOUR MONTH.
```

Writing procedures for these probes is fairly easy. The logic is the same as that used when asking classmates.

Ask a question of a friend.
Check to see which category the answer fits.
Add a tally mark to that category.
Thank your friend.
Ask a question of someone else.

The skeleton program would include the same parts.

```
     TO SURVEY
     QUESTION
     CHECK   ; includes thanks
     SURVEY
     END

     TO QUESTION
     PRINT [IN WHICH MONTH WERE YOU BORN?]
     PRINT [PLEASE TYPE IN YOUR MONTH.]
     MAKE "MONTH FIRST REQUEST
     END

     TO CHECK
     IF :MONTH = "JANUARY MAKE "JAN :JAN + 1
THANKS STOP
     IF  :MONTH = "FEBRUARY MAKE "FEB :FEB +
1 THANKS STOP
     ......
     ......
     IF :MONTH = "DECEMBER MAKE "DEC :DEC  +
1 THANKS STOP
     PRINT  [ARE  YOU  SURE YOU TYPED  IN  A
MONTH?]
     PRINT [CHECK YOUR SPELLING]
     END

     TO THANKS
     PRINT  [  THANK YOU FOR  ANSWERING  THE
SURVEY.]
     BIRTHDAY.CAKE  ;  a turtle drawing of a
birthday cake
     END
```

Before starting the survey, all the values for the months need to be set at 0. This initialization process takes place before SURVEY starts by typing SETUP.

```
TO SETUP
MAKE "JAN 0
MAKE "FEB 0
....
....
MAKE "DEC 0
END
```

When the survey is finished, check the totals by printing out the names of the variables.

```
PO NAMES
"JAN IS 10
"FEB IS 8
....
....
....
"DEC IS 12
```

Save the results of the survey by saving the workspace in a file which shows the date or group.

```
SAVE "BDAY.SMITH.CLASS
SAVE "BDAY.2.10.85
```

You can use the same survey another day in another class or with another group of people. At the end of two weeks you have information from several sources.

MAKE IT BETTER

The survey program is only a skeleton which has many ways to be improved. The way QUESTION and THANKS looks on the screen can be changed with CURSOR and better spacing. A picture, a pause, and more information are all good improvements.

The programming also has much that can be improved. The skeleton program has not used list processing. Instead of checking the :MONTH twelve different times, you could check it against a list of months.

```
     MAKE "MONTHS [JANUARY FEBRUARY MARCH
APRIL MAY JUNE JULY AUGUST SEPTEMBER OCTOBER
NOVEMBER DECEMBER]
```

SETUP.LIST starts each month out with an empty list.

```
TO SETUP.LIST
MAKE "JANUARY []
MAKE "FEBRUARY []
....
MAKE "DECEMBER []
END
```

Tipps continued

Now the BETTER.SURVEY can use a better way to check the lists.

```
TO BETTER.SURVEY
QUESTION
BETTER.CHECK
BETTER.SURVEY
END

TO BETTER.CHECK
TEST MEMBER? :MONTH :MONTHS
IFTRUE TALLY
IFTRUE THANKS
IFFALSE TRY.AGAIN
END
```

If the month is found, a tally is added to the list which is attached to the month name.

```
TO TALLY
MAKE :MONTH FPUT "X THING :MONTH
END
```

THANKS and TRY.AGAIN are simple or improved variations from the old SURVEY.

The results of the survey can be seen by printing out names from BETTER.SURVEY.

```
PO NAMES
"JANUARY IS [X X X]
"FEBRUARY IS [X X]
...
...
"DECEMBER IS [X X X X]
```

The results can also be saved in a file. It may be handy to ERASE all the procedures and SAVE a file which has only the variables names and values in it.

```
ERASE PROCEDURES
SAVE "MONTHS.SMITH.CLASS
```

HOW MANY APRIL BIRTHDAYS?

The number of birthdays for each month can be printed out with PO NAMES. This display is not very pretty. A nicer way to show the number of birthdays would be with a display on the text screen.

Since SURVEY counted January birthdays with the variable JAN, the value of the variable can be used as an input to REPEAT. To see how many January birthdays are in the group surveyed, just type

```
REPEAT :JAN [ PRINT1 "J ]
```

Creating a display of all the birthday months requires a separate line for each month. Use the cursor to create a display on the TEXTSCREEN.

```
    TO DISPLAY
    NODRAW
    CURSOR 0 10  PRINT1 "JANUARY
    CURSOR  12 10 REPEAT :JAN  [PRINT1  "J]
PRINT []
    PRINT1 "FEBRUARY
    CURSOR  12  11 REPEAT :FEB [PRINT1  "F]
PRINT []
    PRINT1 "MARCH
    CURSOR  12 12 REPEAT :MAR  [PRINT1  "M]
PRINT []
    ....
    ....
    ....
    ....
    PRINT1 "DECEMBER
    CURSOR  12  21 REPEAT :DEC  [PRINT1  "D
PRINT []
    END
```

Calling DISPLAY gives a chart such as this one.

```
JANUARY    JJJ
FEBRUARY   FFFF
MARCH      MM
APRIL      AAAAA
MAY        MMMM
JUNE       JJ
JULY       JJJ
AUGUST     AAAA
SEPTEMBER  SS
OCTOBER    OO
NOVEMBER   NNNN
DECEMBER   DDD
```

BETTER.SURVEY collects X'S for each answer; this provides an easier way to write a display procedure. BETTER.SURVEY uses recursion to work through the list of MONTHS and set the LINE for the display.

```
TO BETTER.DISPLAY :MONTHS :LINE
IF :MONTHS = [] STOP
CURSOR 0 :LINE
PRINT1 FIRST :MONTHS
CURSOR 12 :LINE
PRINT THING FIRST :MONTHS
BETTER.DISPLAY BUTFIRST :MONTHS :LINE + 1
END
```

BETTER.DISPLAY shows an X for each birthday from BETTER.SURVEY.

Tipps continued

JANUARY	X X X
FEBRUARY	X X X X
MARCH	X X
APRIL	X X X X X
MAY	X X X X
JUNE	X X
JULY	X X X
AUGUST	X X X X
SEPTEMBER	X X
OCTOBER	X X
NOVEMBER	X X X X
DECEMBER	X X X

BEYOND THE SURVEY

Try making a survey to collect information. The topic of the survey and the sophistication of the programming are adjustable to any level. If you find that students need a Logo tool which they cannot invent for themselves, you can provide a skeleton program as a hint for elaboration. After you have them hooked on surveys, it is time for another look at how to organize and compare the results. That will be the topic for next month's column.

Steve Tipps is the West Professor of Education at Midwestern State University in Wichita Falls, TX. He presents Logo workshops for school systems throughout the United States.

Terrapin Utilities II Disc Announced

Terrapin announces the release of the Utilities II Disk for Terrapin Logo. The disk contains over two dozen files for application in many areas of Logo use. Included are programs to print pictures from Terrapin Logo with the Apple Imagewriter, drive the Apple Color Plotter, load and save KoalaPad pictures (it works with Power Pad pictures, too), draw low resolution graphics, write text characters on the Logo graphics screen, and a translator to run Apple Logo files. Although Terrapin versions 1.3 and 2.0 are recommended for the Utilities II disk, a USEFUL.PROCS file is furnished so that earlier versions may be used. The disk is accompanied by excellent documentation. To order, send $20 to Terrapin, Inc., 222 Third Street, Cambridge, MA 02142, or call (617) 492-8816.

NLXionary

A Lectionary of Selected Logo Readings

Commentary by Robbie and Griff Wigley

"To Sprite or not to Sprite?" by Jim Muller, Turtle News, Young Peoples' Logo Association, March 1983.

>We resurrect this two year old article from the archives because it seems the sprite vs. turtle issue is still current. Several versions of Logo on the market implement sprites or give turtles sprite features. Teachers must make decisions on how to approach them.

>Jim Muller addresses the article mainly to Apple owners who wonder whether it's worth the extra cost to add a sprite board to their computer. He adds that others must concern themselves with how to use their sprites if they are provided in their Logo version. "Electronic deficiencies" of sprite implementation are a source of irritation to Muller. For example, the "no more than four in a row" limitation, being unable to draw (TI Logo), lack of shape rotation as orientation changes, and sprite commands different from those of the turtle. He contends that these machine limitations prevent the "no threshold, no ceiling" experience so critical to Logo.

TWO BY TWO

>He compares the work of two students, one a virtuoso of sprite flashiness without any creative thinking, the other a visualizer of results while developing turtle procedures without reference to the display screen. He wonders which has progressed at his/her own level of understanding; which has progressed from one dimensional to multi dimensional thinking; which has come to see some of the power of imagination. He cites examples of students who, "after learning that virtually anything can be done in turtle graphics, ...have become frustrated with the limits imposed by sprites. It is like once tasting the excitement and strategic limitlessness of chess, you are told you must play only checkers."

>In answer to those complaining of boredom with turtle graphics, Muller suggests versions of Logo with multiple tur-

NLXionary continued

tles, such as Radio Shack's Color Logo. He sees a smooth transition between single and multiple turtles, as opposed to the "total break in symmetry for the purpose of animation" with sprites. Each turtle can draw and be assigned a different shape that will automatically turn in the direction assigned. He believes these features are legitimate extensions of the original concepts presented in initial implementations of Logo. And he sensitively worries about physically and developmentally disabled students who, having tasted success with turtle graphics, must contend with machine limitations and the confusing features of sprites. "...what problems might arise from having to explain why the computer won't do what they want it to do? ...the need for symmetry becomes crystal clear."

SYMMETRY IS IMPORTANT

>We agree with Muller that the additional features of a Logo version such as Color Logo extend traditional turtle graphics in a consistent fashion. The "symmetry," as he puts it, is clearly a desirable feature that would eliminate some of the confusion as a student moves to more varied uses of the language. However, we can't see how telling a student that a computer can't do what he or she wants is going to cause them major problems. Muller nearly becomes patronizing of students with learning difficulties at this point. Students could be told that the computer can't handle their good ideas yet. Finding a creative way around a computer's limitations can even be exhilarating.

>While multiple turtles are a desirable extension for Logo versions, Muller must admit that some students will become immediate mode virtuosos of these Logo versions, too. Most students (and even some teachers!) treat potent learning tools in trivial ways at times. When a student spends session after session designing shapes for the multiple turtles to "carry," yet doesn't progress to building procedures, will Muller blame Radio Shack? We think it better to assess the needs and abilities of the student, as well as the type of instruction he or she is receiving than to blame the hardware.

A PERSONAL PERSPECTIVE

>We both have taught Logo for nearly three years. Much of our early teaching was in camp and learning center settings where students' learning was of the short and intense variety during non-school hours. For the last year and a half, we have taught Logo at a small independent K-4 school during the school day. The overall student / computer ratio is 4 to 1, with computers both in a lab and in individual classrooms. The major mistakes with our early teaching was our tendency to hurry...the "got to give 'em their money's worth" kind of approach. And we immediately concurred with those who said, "Don't start them out with sprites - they're so flashy and fun that they'll never learn the important programming concepts." But now the pressure to hurry (real or imagined) is off. Our present students will be around for a while. Learning is more individually paced. And we think differently now about sprites.

>Until we introduced the sprites, we were forever using "turtle activities" for the younger group to keep them interested: mazes, shoot-at-the-targets, copy-the-drawing, play turtle, and whatnot. In our minds, we were doing it to prepare them better for "the real Logo world." It has since dawned on us that education has always done that number on students, and here we were doing it in the name of Logo.

SPRITES BEFORE TURTLES?

>Sprites seem to be a better environment than the turtle (instant or otherwise) to explore for many first and second grade students. Colors, shapes, directionality, and movement are all a big deal to these students. A sprite environment easily lends itself to exploration and creation without having to devise endless "props" to keep them interested. They are concrete and fun, machine limitations not withstanding.

>To us, third and fourth graders generally seem most ready for the full Logo language, though we wouldn't deny that some children can dive in with sustained interest prior to that. We do think that it's easier to teach many fundamental Logo programming concepts with turtle graphics. But many students are more motivated to put those concepts to use with sprites than with the turtle. We coach them in their work with both.

>As the computer technology develops, hardware limitations and distinctions between sprites and turtles will fade. Soon, every turtle will be a dynamic turtle, with touch, speech, and voice recognition capabilities. But questions of how to teach what and why will remain. We, too, need environments to explore.

NLXionary continued

(Authors' Note: We acknowledge that much has changed in the two years since Muller's article was written. We invite a response from the author and other NLX readers. Our address is 918 College Street, Northfield, MN 55057.)

Robbie and Griff Wigley are co-owners of Family Computing, Inc. Robbie teaches Logo to grades K - 4 in Prairie Creek Community School in Northfield, MN. Griff is a project facilitator in the Faribault Public Schools, and conducts Logo workshops for teachers throughout the state.

TurtleTips

by
Donna Lanyi
and Jane Toth

Have you discovered the creative students in your classroom? Some of these students may be unexpected shining stars at the computer. Good for them! Students' abilities and talents with turtle graphics are growing quickly after five months of school. In order to meet their needs, you need to be prepared to work with students at various levels (a Logo understatement!). We have found two ways to help students at both ends of the spectrum: task cards and the PRINT command.

TEACHER HINTS

There are always a few students who are unable to be productive at the computer using the discovery methods of Logo. These students need guidance on what to try next. Using task cards helps to give these students a base from which to work. To assist these students, we have developed task cards which contain certain Logo concepts.

Each card has the students try a few tasks, record what they see, and then use these ideas to make a picture or project of their own. Task cards can also be used with students at the top end of the spectrum in order for them to experience new concepts.

If you do not want to develop your own Logo task cards, see the "Teacher to Teacher" column in the October 1984 issue of the NLX for a review of a half-dozen sets which are available commercially. (Ed. Note: In addition, EduComp has just published sets of cards for primary and elementary use. EduComp, 14242 Wyeth Avenue, Irvine, CA 92714.)

For students who want to go beyond turtle graphics skills, introduce the PRINT command. PRINT makes it possible for students to write poetry, stories, letters, jokes, spelling words, or vocabulary words. It opens up a whole new world of Logo to your students. The SETCURSOR (or CURSOR) command adds the pizzazz, as Cathy Frank shows in her Logo wordles article on page 1 of this issue!

PROCEDURE OF THE MONTH

HAPPY VALENTINE'S DAY! Encourage students to create valentines with Logo. Hearts are not easy to draw, but are fun to try. Use the NLX ABC's to write a valentine message on the drawing. Print the valentine drawing out on paper, if possible. If you are interested in a free listing of a sample heart procedure, please send us a self addressed stamped envelope.

We invite elementary teachers and students to send procedures or ideas to be shared. Donna Lanyi and Jane Toth, Beall Avenue School, 716 Beall Avenue, Wooster, OH 44691.

Donna Lanyi and Jane Toth use Logo in their elementary classrooms in the Wooster (OH) City Schools.

Logo Notes

Microquests, a newsletter featuring a fully developed monthly Logo adventure, promises many stimulating ideas for projects. Powerful ideas and concepts from science and mathematics are presented and discussed in each issue. Published by Kathleen Martin and Donna Bearden, well known Logo authors, Microquests is available for $25 per year (September through May). Write to Microquests, PO Box 337, Grapevine, TX 76051.

The National Council of Teachers of Mathematics (NCTM) announces a Logo design contest in conjunction with Mathematics Education Week activities for April 14 - 20, 1985. For more information, write NCTM, 1906 Association Drive, Reston, VA 22091, or call (703) 620-9840.

Listful Thinking

by
Glen Bull and Paula Cochran

Power Tools

Did you ever have four or five PRINT statements in a row? Would you like a way to print five lines of text without putting PRINT in front of each line? You'll learn how in this month's column, if you use the WRITE procedure. It's great for creating adventure stories, for example. This month we're also going to look at two other list tools, SELECT and CREATE.

THE TEXT COMMAND
(Or "Wow! What Is All that Stuff in There?")

For nearly a year we haven't used a graphics command in this column. That isn't because lists aren't useful in graphics applications. On the contrary, they are very useful for graphics. (Tom Lough notes that a list of turtle commands can correspond directly to the addition of a series of vectors in physics.) However, we're trying to help dispell the myth that Logo is only a graphics language. Now we're going to ask you to type in a graphics procedure, but only so that we can look at its text. First define this procedure.

```
TO HAT
SETH 90
FORWARD 50 BACK 40
LEFT 90
REPEAT 4 [FORWARD 30 RIGHT 90]
END
```

Then, in the immediate mode, look at the text of the HAT procedure, using the TEXT command.

```
?PRINT TEXT "HAT
[] [SETH 90] [FORWARD 50 BACK 40] [LEFT
90 [REPEAT 4 [FORWARD 30 RIGHT 90]
```

When we look at the text of the procedure, it seems to be a list of commands. But what is the empty list [] at the beginning? To find out, let's make a new procedure that has an input. This new procedure, called CAP, lets us make a hat of any size.

```
TO CAP :SIZE
SETH 90
FORWARD :SIZE BACK :SIZE * ( 4 / 5)
LEFT 90
REPEAT 4 [FD :SIZE * (3 / 5) RT 90]
END
```

```
?PRINT TEXT "CAP
[:SIZE] [SETH 90] [FORWARD :SIZE BACK :S
IZE * ( 4 / 5 )] [LEFT 90] [REPEAT 4 [FD
:SIZE * ( 3 / 5 ) RT 90]]
```

The mystery of the empty brackets at the beginning is solved. That's where the inputs (such as :SIZE) are listed if there are any. If there are no inputs to the procedure, empty brackets appear at the beginning.

THE WRITE COMMAND

So far we appear to have found a bizarre way to look at the text of a procedure. But what if we wrote a procedure that contained English sentences instead of Logo commands? As you type this in, be sure to press the RETURN key at the end of each line shown below.

```
TO MYSTERY
SUNLIGHT DAPPLED THE FOREST FLOOR.
JOHNATHAN SLOWLY AWOKE, AND STRETCHED.
HE COULD FEEL THE ROUGH BARK OF AN OAK
AGAINST HIS BACK. THROUGH THE LEAVES
HE GLIMPSED A PEAKED CAP ATOP A SMALL
FIGURE.
END
```

After this MYSTERY procedure is defined, we can print the text in it using the TEXT command. However, what a mess!

```
?PRINT TEXT "MYSTERY
[] [SUNLIGHT DAPPLED THE FOREST FLOOR.]
[JOHNATHAN SLOWLY AWOKE, AND STRETCHED.]
 [HE COULD FEEL THE ROUGH BARK OF AN OAK
] [AGAINST HIS BACK. THROUGH THE LEAVES]
[HE GLIMPSED A PEAKED CAP ATOP A SMALL]
[FIGURE.]
```

We can get rid of the empty brackets at the beginning by printing everything but the first list in the MYSTERY procedure.

```
?PRINT BUTFIRST TEXT "MYSTERY
[SUNLIGHT DAPPLED THE FOREST FLOOR.] [JO
HNATHAN SLOWLY AWOKE, AND STRETCHED.] [H
E COULD FEEL THE ROUGH BARK OF AN OAK] [
AGAINST HIS BACK. THROUGH THE LEAVES] [H
E GLIMPSED A PEAKED CAP ATOP A SMALL] [F
IGURE.]
```

Listful Thinking continued

It still looks like a bit of a mess. However, there is a procedure from an earlier Listful Thinking column which can come to the rescue. The list tool INSPECT was developed to print out lists in neat column. You will find the INSPECT procedure at the end of this column. If you don't already have INSPECT in your toolbox, type it in now. Here's how to use INSPECT.

```
?INSPECT BUTFIRST TEXT "MYSTERY
SUNLIGHT DAPPLED THE FOREST FLOOR.
JOHNATHAN SLOWLY AWOKE, AND STRETCHED.
HE COULD FEEL THE ROUGH BARK OF AN OAK
AGAINST HIS BACK. THROUGH THE LEAVES
HE GLIMPSED A PEAKED CAP ATOP A SMALL
FIGURE.
```

This is the basis for a procedure to WRITE the text.

```
TO WRITE :PROCEDURE
INSPECT BUTFIRST TEXT :PROCEDURE
END
```

Here's how WRITE is used.

```
?WRITE "MYSTERY
SUNLIGHT DAPPLED THE FOREST FLOOR.
JOHNATHAN SLOWLY AWOKE, AND STRETCHED.
HE COULD FEEL THE ROUGH BARK OF AN OAK
AGAINST HIS BACK. THROUGH THE LEAVES
HE GLIMPSED A PEAKED CAP ATOP A SMALL
FIGURE.
```

Compare the text in a procedure designed for use with WRITE and a procedure consisting of a series of PRINT statements.

```
TO MYSTERY
SUNLIGHT DAPPLED THE FOREST FLOOR.
JOHNATHAN SLOWLY AWOKE, AND STRETCHED.
HE COULD FEEL THE ROUGH BARK OF AN OAK
AGAINST HIS BACK. THROUGH THE LEAVES
HE GLIMPSED A PEAKED CAP ATOP A SMALL
FIGURE.
END
TO MYSTERY.II
PRINT [SUNLIGHT DAPPLED THE FOREST FLO!
OR.]
PRINT [JOHNATHAN SLOWLY AWOKE, AND STR!
ETCHED.]
PRINT [HE COULD FEEL THE ROUGH BARK OF!
 AN OAK]
PRINT [AGAINST HIS BACK. THROUGH THE L!
EAVES]
PRINT [HE GLIMPSED A PEAKED CAP ATOP A!
 SMALL]
PRINT [FIGURE.]
END
```

When you use WRITE you save time (by not typing PRINT over and over). It's also easier to see where the end of each line will fall.

<u>SELECTING STUDENTS</u>

In a previous column, we developed a list tool called PICK for selecting items randomly from a list. For example, you could use it to pick students.

```
?MAKE "CLASS [TOM SUE JEFF SAM SALLY]
?PRINT :CLASS
TOM SUE JEFF SAM SALLY
?PRINT PICK :CLASS
JEFF
```

The list is not changed when a student is picked.

```
?PRINT :CLASS
TOM SUE JEFF SAM SALLY
```

Over the long term, PICK should choose each student about the same number of times. However, in the short term some students may be chosen more often than others. What we want is a means of selection without replacement. In other words, if we put all the names in a hat and pulled one out, we would not put it back in the hat again.

The DELETE procedure from last month can be combined with PICK to make a procedure which SELECTs without replacement. Listings for PICK and DELETE appear at the end of the column. Type them in now if you don't already have them.

```
TO SELECT :LIST
MAKE "ITEM PICK THING :LIST
DELETE :ITEM :LIST
OUTPUT :ITEM
END
```

Here's how to use SELECT. Note that, as input to SELECT, we use the <u>name</u> of the list as input (with ") instead of the <u>value</u> of the list (with :), so that the globally assigned values are deleted. (Ed. Note: See also Jim McCauley's related remarks in the Q & A column in this issue.)

```
?MAKE "CLASS [TOM SUE JEFF SAM SALLY]
?PRINT :CLASS
TOM SUE JEFF SAM SALLY
?PRINT SELECT "CLASS
SAM
?PRINT :CLASS
TOM SUE JEFF SALLY
```

The way in which SELECT differs from PICK is that SELECT randomly chooses an item from a list <u>and</u> deletes it from the list.

Listful Thinking continued

SELECT gradually destroys the list as it randomly selects and deletes. We need a way to recreate the list without typing it in again. Let's start by making a procedure that contains a list of class names. This may seem odd, but by now we hope we've built up some credit.

```
TO CLASS.ROLL
TOM KELLER
SUE MORGAN
KYSER LOUGH
SAM JOHNSON
SALLY ALLYSON
END
```

Use the TEXT command to look at the students in the procedure.

```
?PRINT TEXT "CLASS.ROLL
[] [TOM KELLER] [SUE MORGAN] [KYSER LOUG
H] [SAM JOHNSON] [SALLY ALLYSON]
```

Now let's make a list from the text of the CLASS.ROLL procedure. Here's how to create a list from a such a procedure.

```
TO CREATE :LIST.NAME :PROCEDURE
MAKE :LIST.NAME BF TEXT :PROCEDURE
END
```

Using CREATE, we can make a list out of the CLASS.ROLL.

```
?CREATE "CLASS "CLASS.ROLL
?PRINT :CLASS
[TOM KELLER] [SUE MORGAN] [KYSER LOUGH]
[SAM JOHNSON] [SALLY ALLYSON]
```

Now we can CREATE a new list any time we want. It doesn't matter if SELECT destroys the list of students.

```
?PRINT SELECT "CLASS
SUE MORGAN
?PRINT :CLASS
[TOM KELLER] [KYSER LOUGH] [SAM JOHNSON]
[SALLY ALLYSON]
```

POWER TOOLS

This month we have presented three new tools: WRITE, SELECT, and CREATE. As you experiment with them, we hope that you will discover how powerful these tools can be (yes, power tools!). Although they won't chop down any cherry trees, they can be combined to build exciting projects for your students. Next month, we'll show you how to use your new power tools to build a STORY WRITER.

PROCEDURES

Here are Apple Logo and IBM Logo procedures.

```
TO INSPECT :LIST
IF EMPTYP :LIST [STOP]
PRINT FIRST :LIST
INSPECT BUTFIRST :LIST
END

TO PICK :LIST
OP ITEM 1 + (RANDOM COUNT :LIST) :LIST
END

TO DELETE :ITEM :LIST
MAKE :LIST DELETE.LOOP :ITEM THING :LIST []
END

TO DELETE.LOOP :ITEM :OLDLIST :NEWLIST
TEST EMPTYP :OLDLIST
IFTRUE [PRINT SE :ITEM [NOT FOUND]]
IFTRUE [OUTPUT :NEWLIST]
TEST :ITEM = FIRST :OLDLIST
IFTRUE [OP SE :NEWLIST BF :OLDLIST]
OP DELETE.LOOP :ITEM BF :OLDLIST LPUT
       FIRST :OLDLIST :NEWLIST
END
```

Here are Terrapin Logo procedures for the Apple and Commodore.

```
TO INSPECT :LIST
IF :LIST = [] STOP
PRINT FIRST :LIST
INSPECT BUTFIRST :LIST
END

TO PICK :LIST
OP ITEM 1 + (RANDOM COUNT :LIST) :LIST
END
```

(In Terrapin Version 1, PICK also requires these procedures: ITEM, ITEM.LOOP, COUNT, and COUNT.LOOP.)

```
TO ITEM :NUMBER :LIST
OUTPUT ITEM.LOOP :NUMBER :LIST 1
END

TO ITEM.LOOP :NUMBER :LIST :COUNT
IF :LIST = [] OP (SE [THERE ARE ONLY] :COUNT
       [ITEMS IN THIS LIST.] )
IF :NUMBER = :COUNT OP FIRST :LIST
OP ITEM.LOOP :NUMBER BF :LIST :COUNT + 1
END

TO COUNT :LIST
OUTPUT COUNT.LOOP :LIST 0
END
```

Listful Thinking continued

```
TO COUNT.LOOP :LIST :NUMBER
IF :LIST = [ ] OUTPUT :NUMBER
OP COUNT.LOOP BF :LIST :NUMBER + 1
END

TO DELETE :ITEM :LIST
MAKE :LIST DELETE.LOOP :ITEM THING :LIST [ ]
END

TO DELETE.LOOP :ITEM :OLDLIST :NEWLIST
TEST :OLDLIST = []
IFTRUE PRINT SE :ITEM [NOT FOUND]
IFTRUE OUTPUT :NEWLIST
TEST :ITEM = FIRST :OLDLIST
IFTRUE OP SE :NEWLIST BF :OLDLIST
OP DELETE.LOOP :ITEM BF :OLDLIST LPUT FIRST
        :OLDLIST :NEWLIST
END
```

Glen Bull is a professor in the University of Virginia's School of Education, and teaches Logo courses at both the graduate and undergraduate level. Paula Cochran is a reformed English major who studied linguistics at the University of Cambridge. She is now a speech-language pathologist working with language-disabled children.

Logo Disserts:

Dissertations Dealing with Logo

by Barbara Elias

A Logo learning experience was used to explore experimentally the effects of programming on the conditional reasoning ability of fifth grade children. Conditional reasoning statements incorporate the "if - then" connective and commonly occur in our daily communication. Conditional branch statements appear in most high level computer languages available today. The statements (instructions) change the natural way in which computer programs are usually executed. The changes which occur depend on certain conditions being satisfied.

THE STUDY

Fifth graders, in a randomly assigned experimental group, learned to use the conditional branch statement in Logo in individual projects they worked on during a 15 week treatment period. About 2 hours per week were spent with teachers who used a guided problem centered approach to instruction. The need for conditional branch statements grew out of programming projects which involved game playing, question asking / answering procedures, and recursive procedures. A total of 42 children participated as randomly assigned experimental or control subjects. Control group children were not involved with Logo.

RESULTS

A traditional scoring of tests given at the end of the study showed no significant difference in scores of the two groups. An additional scoring procedure, however, indicated significantly better performance of the experimental group on one of the conditional reasoning principles. One other finding from the study of interest for further investigations: reading achievement scores for the control group showed a significant increase during this period!

>>>Seidman, R. H. (1981). The Effects of Learning the Logo Computer Programming Language on Conditional Reasoning in School Children (Doctoral dissertation, Syracuse University, 1980). Dissertation Abstracts International, 41, 2249B.

(Ed. Note: Each month, Barbara Elias highlights a dissertation or thesis dealing with Logo. She is accepting copies of recent research results for the NLX Dissertation and Thesis Repository. Reports may be mailed to her in care of NLX, PO Box 5341, Charlottesville, VA 22905.)

Barbara Elias is an assistant professor in the Education Department of Virginia State University in Petersburg, VA, and is a doctoral candidate at the University of Virginia.

Graphics Dump From Logo

The UniPrint Parallel Printer Interface for the Apple II+ and IIe computers is compatible with Apple Logo, and permits a graphics dump from Logo to more than 25 different popular parallel printers. The modest price of $89 includes the interface card, Centronics cable, and instructions. Write to Videx, 1105 NE Circle Boulevard, Corvallis, OR 97330, or call (503) 758-0521.

Teacher to Teacher

Logo, Words, and Ideas Teacher Training Materials

reviewed by
James Fry

Logo, Words, and Ideas, by Thomas Boe, is a set of teacher training materials and lesson plans from the Minnesota Educational Computing Corporation (MECC) for a five session Logo course for Apple Logo teachers working with students in grades 5 - 12, computer coordinators and resource persons, and teachers of language arts, science, math, and computer science who are interested in using Apple Logo in their curriculum areas.

The introductory pages give a clear explanation of the goals, prerequisite skills, class format and setting (an Apple lab is recommended, with no more than two participants per computer), organization of the materials, and the preparation of participant diskettes. I welcome this information at the beginning of the manual.

The goals of the training activities are designed to help teachers:

-challenge students who are ready to go beyond turtle graphics,

-integrate Logo into language arts, science, and math,

-encourage both top down design and bottom up exploration.

The materials are designed for five sessions of at least two hours each. It is also suggested that the class sessions be spaced at least a week apart, to allow the participants time to experiment on their own.

ON THE INSIDE

The manual has five sections. The first section contains instructor notes, including helpful scripts for Logo lessons, and suggestions pertaining to materials, time, objectives, preparation, hands-on activities, and follow-up activities. This section is useful for both experienced and inexperienced Logo instructors.

The second section is a set of forty transparency masters of procedures and command explanations. The large print produces clear easy-to-read masters which give commands and their meanings and explanations, suggested activities, sample procedures, and drawings of what the procedures produce on the computer screen.

A set of participant activity pages makes up the third section, which contains explanations of each lesson and extended activities which coordinate well with the instructor notes.

The fourth section is devoted to participant notes; copies of transparencies for each lesson are provided, with space for personal notes.

The final section contains reference materials, including a summary of Apple Logo commands and word and list primitives. A bibliography and an index of word and list primitives round out this section.

THE FIVE SESSIONS

The five sessions focus on:

* recursion - investigating tail and embedded recursion and developing a theory to explain it
* introduction to Logo words - using word and list primitives
* singluar to plural - applying word and list commands to a group problem solving activity and identifying the steps in creating a computer solution to a problem
* hurkle project - participating in a Logo project activity which provides a variety of choices to explore
* mathematics and science - working with coordinate geometry and absolute angular directions and applying angular and coordinate commands to applications in mathematics and science.

IN SUMMARY

MECC's Logo, Words, and Ideas contains specific information, objectives, and support materials for both experienced and inexperienced Logo teachers. Although there is structure in the materials, creative exploration and discovery in the lessons is encouraged.

It is available for $49 from MECC, 3490 Lexington Avenue North, St. Paul, MN 55112, or call (612) 481-3527.

Teacher to Teacher continued

Logo and Mathematics Workbook Elementary Grade Materials

One interesting idea to come out of the Logo 84 conference is the Logo and Mathematics Workbook written by Reinhold Wappler of the New Canaan (CT) Country School. Wappler shared copies with colleagues at the conference, and showed that it was designed to be used as off-computer practice of what is learned in the school computer labs.

He described the book as having "emerged from three years of teaching mathematics via Logo in grades 2 through 5." And don't let the word "workbook" in the title mislead you; the book contains a reasonable balance of directed activities and suggested extensions, all related to specific mathematics concepts. Included are sections on lines, arcs, circle drawing commands, regular polygons, procedures, x- and y-coordinates, headings, and variables. In addition, class or group projects on string art, nested figures, and patterns of numbers and variables are suggested. The book is conveniently set up for use with any version of Logo.

SETHEADING :MATH

Those of us who have worked with younger students often find that, if left undirected, they learn very little from Logo and may lose interest quickly. The exercises in the book are all based on particular mathematical ideas appropriate for second through fifth graders. For example, Logo is easily adapted for the exploration of mathematical concepts such as external angles, coordinates, decimals, and number lines.

A helpful addition to this part of the book would be scripts for the teacher or suggestions for the computer activities which precede the appropriate workbook pages.

One of the powerful ideas in algebra, trigonometry, and calculus is that a change in the value of one variable produces a change in the value of another as a consequence. With this idea in mind, Wappler develops activities for turtle heading and position, and how these change as the turtle moves from one place to another.

Especially convenient teaching aids include a transparent turn indicator you line up with the turtle to find the number of degrees for a left or right turn, and a heading circle, used to estimate the current heading of the turtle.

IN SUMMARY

If you are a classroom teacher looking for some practical mathematics ideas and activities to use with Logo, give Wappler's Logo and Mathematics Workbook a try. Its 100 pages provide you with new ways of doing familiar activities.

Logo and Mathematics Workbook is available from the author. Send $13 to Reinhold Wappler, 252 Carter Street, New Canaan, CT 06840. For addresses outside the United States and Canada, enclose $5 extra for air mail postage.

Send your review reactions and comments to Logo Information For Teachers (LIFT), PO Box 5396, Plymouth, MI 48170.

James Fry teaches with Logo at Novi Woods Elementary School in Novi, MI, and is a cofounder of LIFT.

Q and A

by

Jim McCauley

Rather than address a specific question this month, let me deal with a subject of considerable confusion, both to a number of correspondents to this column and many of my students. The subject is the manner in which values are bound to variable names in Logo.

In the December Q & A, I introduced the two methods by which values are associated with names: by global assignment with MAKE and by binding to local variable names on entry to a procedure. I encouraged readers to try to use local bindings to the maximum extent possible, so that they might avoid the confusion presented by mixing local and global bindings. At the time, I didn't include an example of such a confusion; let me do it now. It serves to illustrate several points simultaneously.

Let's begin by creating a global assignment with MAKE:

```
MAKE "X "FRED
```

Now, let's define a simple procedure that accepts no inputs. All it does is print the value bound to X.

```
TO CONFUSING
PRINT :X
END
```

Q and A continued

When we type CONFUSING, the result is

```
FRED
```

However, if we create a superprocedure that calls CONFUSING, we can create a truly interesting mess.

```
TO SUPERCONFUSING :X
PRINT :X
CONFUSING
END
```

Oh, my. How does this procedure behave? The value FRED is already bound globally to the variable X. What happens when we try to associate a new value to the local variable X in SUPERCONFUSING? Furthermore, how does CONFUSING behave when it is called from within SUPERCONFUSING? Does it use the globally assigned value of FRED or does it use the value bound to the variable X in SUPERCONFUSING?

MOVING RIGHT ALONG

Let's end this Logo soap opera. If we type

```
SUPERCONFUSING "STUFF
```

we get in return

```
STUFF
STUFF
```

Oho! So when we pass a value to the variable X in SUPERCONFUSING as we call upon it, the value passed as a parameter (STUFF) supersedes the value that is bound globally (FRED). Furthermore, when CONFUSING is called from within SUPERCONFUSING, it "inherits" the value of X bound on entry to SUPERCONFUSION. In other words, in the versions of Logo that are available on microcomputers today, values that are bound locally take precedence over values that are bound globally.

This has the potential for causing some difficulty. We can see that CONFUSING has two very different behaviors. If it is called from TOPLEVEL, it uses the globally bound value for X. But if it is called from within another procedure which has a different value bound to X, that is the value which is used instead. While this erratic behavior can be usefully exploited in some special circumstances, it has a tendency to trap the unwary.

HOW TO UNDERSTAND?

Perhaps the best way to understand what is going on is to think of it this way. Many Logo primitives and procedures are able to accept inputs and bind them locally to variable names. Why not simply think of Logo's TOPLEVEL as a continually running primitive that can accept values bound to variable names, and that the MAKE primitive is the means by which this is accomplished? (Actually, this isn't too far from the truth.)

You should be aware that other computer languages use very different strategies for establishing the relationships among the local and global variables in use. There may even be some versions of Logo developed that do not follow these binding conventions. As you begin to write more complicated programs, take special care of how you treat your variables.

(Ed. Note: If you have questions about Logo programming techniques or the thinking which leads to Logo programming, send them to NLX Q & A, PO Box 5341, Charlottesville, VA 22905. Each month, Jim McCauley answers selected questions in this column.)

Jim McCauley is a Coordinator of Computer Education for the Santa Clara (CA) County Office of Education, and has written Logo articles for many national publications.

NLXual Challenges

by
Robs Muir

Stand In a Corner

While the cherry trees were blooming in Washington, DC, last December, normally sunny Southern California was slogging through unseasonable winter weather. With some schools even closed due to snow (!), my students and I were shut in during our regular recesses because of the rain.

INDOOR SPORTS

An indoor game that my students (5th and 6th graders) played and really enjoyed that day forms the basis of this month's challenge. It gives us adults an opportunity to experience an example of the integration of classroom experience into the Logo envi-

NLXual Challenges continued

ronment. It also gives us a change to explore one of Piaget's postulates of intelligence.

In a slim book called The Psychology of Intelligence (Littlefield Adams & Co., 1976), Jean Piaget postulates that all problem solving activities involve several distinct stages. He calls the first stage "anticipatory schemata," representing the initial plan of attack for a given problem. This is, according to Piaget, a necessary component of intelligent behavior.

THE GAME

The game is generally called something like "Four Corners," and is played in a room with four designated stations or areas. In our case, we used the corners of the room. Each station is designated by a number, 1 through 4.

A stationmaster is appointed. This persons counts reasonably slowly to ten with closed eyes while everyone else (including the teacher!) scrambles to get to any one of the stations. Then the stationmaster calls one of the station numbers. Whoever happens to be in that station is "out" and has to sit down.

The game resumes with another round, using the players who are still "in." The game ends when one player is left, who becomes the new stationmaster.

The game is a unique "possession" of my students. It was spontaneously introduced by a student,, the rules were modified by our group, and the game has since spread to other classes in the school. (I know it didn't really originate with us, but we still call it "ours.") A spontaneous Logo problem was created when someone suggested, "Hey! Let's write a Logo stationmaster game!"

THE CHALLENGE

So, this month's NLXual Challenge to you is to write a stationmaster program that displays the four stations, "counts" to ten, and then randomly chooses a station. This program should possess all of the (minimal) abilities of a human stationmaster.

Before you begin to code your solution in the version of Logo available to you, check to see whether Piaget is correct; have you already formulated a plan? Do you know what you would like the finished product to look like? You might change your preliminary schemata as you progress, but the existance of such a schemata is a function of your own mental functioning, and will, in all likelihood, be a unique spontaneous creation. Try to preserve the core of your original plan in your final program.

Send your working solution (or even a valiant attempt) along with a self addressed stamped envelope to:

NLXual Challenges
Attn: Robs Muir
1688 Denver Avenue
Claremont, CA 91711

After compiling the contributed solutions, we will select several of the most interesting or unusual and redistribute them to each contributor. I promise to include several solutions already written by some of my students, too.

(Ed. Note: If any NLX readers have favorite Logo problems or challenges to share, include them with your solutions.)

Robs Muir is a teacher in the Claremont CA Unified School District and Claremont Graduate School. He presents Logo workshops in the southern California area.

The National Logo Exchange, copyright (c) 1985 by Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly, September through May, $25 (US) per year, mailed first class from Charlottesville, VA. $5 (US) additional per year for addresses outside the USA, Canada, and Mexico. The opinions expressed by the authors are not necessarily those of The National Logo Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress Street, Salem, MA 01971. Address other correspondence to: The National Logo Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor....................................Tom Lough

THE NATIONAL
LOGO EXCHANGE

Volume 3 Number 7 FORWARD 100! March 1985

Up the Logo Ramp to List Operations

by

Tony Stavely

One of the things that charmed me when I first learned about Logo was the slogan, "No threshold, no ceiling." The more I thought about it, the more clear it seemed that the ideal learning situation would be just like a wedge. With no threshold, the learning begins immediately. An easy staircase would have fairly low risers and fairly wide treads. The limiting case is a ramp, an inclined plane gently rising forever.

However, learning Logo isn't exactly like strolling up an endless inclined plane. Exciting, frustrating sometimes, usually fun, but not easy. Furthermore, for many learners, passage into the domain of list processing means climbing a step with a pretty tall riser. What follows is some of my experience getting into that domain.

A FIRST STEP

My first challenge was to write a procedure to raise a number to a power. Terrapin Logo for the Apple has no such primitive, and I thought it would be a good first task after a dozen hours of making pleasing turtle designs. With the help of Bruce Metzger, a colleague who knew LISP, I eventually got the procedure TO RAISE :NUMBER :POWER so that it worked.

That project took more than a week and, although it did not contain list processing as such, it served as a kind of breakaway from turtle graphics. Gradually and steadily, with Bruce's help and, later, with the support of new friends Molly and Dan Watt and the Monadnock Logo Users Group that we founded, I kept posing little challenges to myself. Examples included finding the sum of a list of numbers, inserting an item at a specified location in a list, and creating a game that learned the definitions of words from the user and asked about words it didn't know yet.

LOGO BREAKTHROUGH!

For me, the important breakthrough in mastering list processing happened while I was studying the humble LENGTH procedure in Harold Abelson's Logo for the Apple II (Byte/McGraw-Hill). LENGTH contained a key which helped me understand more of the power of lists. It turns out that whether you want to find the length of a list or the sum of a list of numbers, to reverse the order of letters in a word or words in a sentence, or do anything else with a set of Logo objects, there seems to be one fundamental format.

A typical procedure looks like this:

```
TO DISPLAY :ALIST
IF :ALIST = [] STOP
PRINT FIRST :ALIST
DISPLAY BUTFIRST :ALIST
END
```

This is the format:

Part 1: procedure name
Part 2: variable names
Part 3: STOP rule
Part 4: what to do with the FIRST of a list that is the current value of a local variable
Part 5: recursive call, passing along the BUTFIRST of that local variable's list
Part 6: END

Now study the ADDUP procedure listed below. ADDUP finds the sum of a list of numbers. Do you see the format in the procedure? Note the similarities between it and the DISPLAY procedure above.

```
     TO ADDUP :ALIST
     IF :ALIST = [] OUTPUT 0
     OUTPUT (FIRST :ALIST) + ADDUP  BUTFIRST
:ALIST
     END
```

Logo Ramp continued

Unlike DISPLAY, ADDUP uses OUTPUT rather than STOP to halt things. This may be unfamiliar to some readers. All it means is that each ADDUP called upon gives (OUTPUTs) something to the procedure or command that calls upon it. The last ADDUP in the recursive sequence (the one that receives the BUTFIRST of a one-item list) OUTPUTs a zero. The next-to-last ADDUP OUTPUTs the sum of zero and the FIRST of its :ALIST (the last number of the original list). That sum is OUTPUT to the next-to-next-to-last ADDUP, to be added to the FIRST of its :ALIST, and so on , until the beginning ADDUP completes its addition task and OUTPUTs the complete sum to its caller.

ABOUT RECURSION

Recursion like this is a matter of knowing how to do just a few things and delegating everything you can't do to a helper. The helper turns out to be a copy of yourself. ADDUP knows the sum of an empty list (zero) and it knows how to add the first number of its list to the sum of the remaining numbers. However, it trusts its helper to take care of that part. While the helper is doing its job, the unfinished addition is stored in a "stack" inside Logo, waiting to be completed. If its helper calls a helper, the first helper's addition is stacked, too.

Once an empty list is reached, there are no more helpers needed; the unfinished arithmetic gets taken from the stack one number at a time. Readers may recall "stacking up" breakfast preparation while a family member cleans a skillet or runs next door for another couple of eggs. These recursive procedures do a similar thing. (Ed. Note: For additional OUTPUT ideas, see also "Q and A," NLX November 1984.)

TELL ME ABOUT IT

To share more aspects of Logo list processing and some of its uses in creating microworlds and personal tools, I am working on a book entitled Logo Tools: List Processing as a Powerful Resource. I would love to hear from NLX readers about experiences with list processing. What listful bugs creep into your work and build nests? What kinds of tasks would you like to do with Logo list processing? What projects do your students want to work on? With your assistance, Logo Tools may bridge the gap some learners find at the place where list work begins on the endless inclined plane of Logo.

Send your stories to Tony Stavely, University of Michigan Professional Development Office, School of Education, Ann Arbor, MI 48104. After May 30, send them to me at Keene State College, Keene, NH 03431. Thanks!

Tony Stavely is a Professor of Psychology at Keene State College in New Hampshire, and a faculty member of The Logo Institute. He recently completed two years as co-editor of the NH Association for Computer Education newsletter and is completing a six-month stay at the University of Michigan.

From the Editor

Logo on the Move

Recently, I was privileged to present a Logo workshop to a very special group made up of high school physics teachers and university physics professors. Most sections of the United States and several parts of Canada were represented. Before things got going, I asked the participants to share a little information about themselves and what they knew or have heard about Logo.

To my surprise and delight, most of them reported that Logo was already being used in their high schools and universities! In fact, a number of these physics instructors were at the workshop because students were using Logo in math classes at their schools, and they wanted to be ready for them in physics.

From all earlier indications, Logo has not made appreciable headway into the secondary and postsecondary fields to date. (See, for example, "Gradations of Logo," The Computing Teacher, December 1984 / January 1985.) Apparently, Logo's reputation as a child's language has made higher level educators reluctant to investigate its potential, in spite of early encouragement to do so. (See "Logo: Not Just for Kids," Harold Nelson, Microcomputing magazine, March 1982.)

I have always been concerned about this. A tool as powerful as Logo should not be neglected by the upper end of the educational community. Students at the higher levels are equipped with the thought processes necessary to tap the higher capabilities of the language. This is why I am particularly encouraged by the response of the physics teachers in my workshop. They helped me get rid of a case of the "Logo blahs."

NLXTRA

Attention overseas Logo users! Please send information on Logo books, articles, reports, and teaching aids available in your country for inclusion in the bibliography for the Logo 85 conference. Include publisher addresses and prices, where possible. Thank you. Bibliography, PO Box 5341, Charlottesville, VA 22905.

Do you present Logo workshops for teachers? If so, please let us know so we can publicize your activities in the NLX. We also support such presentations with Logo handout materials, upon request. Here is a workshop tip. At a recent Logo workshop, Glen Bull and Paula Cochran distributed to their participants copies of a Logo file disk which could be used by either an IBM PC or an Apple II computer. How did they do it? They initialized one side with the IBM DOS, then punched a hole in the edge and initialized the other side with Apple DOS 3.3! This technique could be used for any combination of two computers whose disk drives can read single-sided disks.

If you are teaching a summer course in Logo, let the NLX help publicize it. Send your course information to NLX, Attn: Summer Courses, PO Box 5341, Charlottesville, VA 22905.

The second edition of The National Logo Exchange's Logo Research Directory is now available. Editor Regina Sapona has done an outstanding job of organizing the more than 60 entries into operational categories, with cross-referencing. Each entry contains a research summary and descriptors, as well as a mailing address for more information. To receive your copy, send $3 to NLX Research Directory, PO Box 5341, Charlottesville, VA 22905. If you are conducting Logo related research and are not listed in the directory, please enclose information about your work and interests.

Terrapin Logo is offered at a price of $65 by Conroy-LaPointe, 12060 SW Garden Place, Portland, OR 97223. (800) 547-1289.

The NLX ABC's have been distributed to thousands of Logo teachers over the past two years. Now let's see what has been done with them! We would like to publish the NLX Book of Logo ABC's. If you or your students have used the NLX ABC procedures for Logo projects, send us a printout or a listing. We are especially interested in projects done with single letters, so they can be grouped in the A section, the B section, and so forth. However, there is also a section for multiple letter projects, too! Maybe you can combine some ideas from Cathy Frank's outstanding "wordle" article in last month's NLX.

The author of each project selected will receive a free copy of the finished book, and will be mentioned by name in the publication, if desired.

Michael Friendly has agreed to be the editor of this project. Send your ABC projects to Michael Friendly, Psychology Department, York University, Downsview, Ontario, Canada, M3J 1P3.

If you have not yet received your listing of the NLX ABC procedures and would like to be involved with this project, send a self addressed stamped envelope to NLX ABC's, PO Box 5341, Charlottesville, VA 22905. Be sure to specify your computer model and Logo version.

Logo Hawaiian this summer! Join Glen Bull, Paula Cochran, and Tom Lough for "Logo in the Classroom," a 3-hour graduate Logo course from the University of Virginia, to be taught at the famed Punahou School in Honolulu, Hawaii, July 8 - 19, 1985. The course concentrates on Logo applications in language arts, science, and mathematics content areas for teachers with Logo experience. For Logo beginners, an outstanding graduate introductory course is offered, taught by Elaine Blitman and Barbara Jamile. Make your plans early for these popular offerings. For more information, call (800) 223-2544. New York residents call (212) 686-5810 collect. Educational Spectrum, Inc., 21 East 40th Street, New York, NY 10016.

Commodore Logo is available for $59 from Sunburst Communications, Room BN, 39 Washington Avenue, Pleasantville, NY 10570. (800) 431-1934.

"Integrating Logo Into the Curriculum," a mini-conference sponsored by ECOO-SigLogo, is scheduled for March 23, 1985, in North York, Ontario. Included in the program are eleven presentations on Logo in various content areas, ranging from special education to science to storywriting. For more information, contact Paul Davidson, Sloane Avenue P. S., 110 Sloane Avenue, North York, Ontario M4A 2B1 Canada.

The fifth annual National Microcomputers in Education Conference is scheduled for March 13-15, 1985, at Arizona State University. Many of the more than 200 scheduled sessions focus on Logo. For more information, write Gary Bitter, College of Education, Payne 216, Arizona State University, Tempe, AZ 85287, or call (602) 965-7363.

The Alberta Teacher's Association Computer Council announces ATACC 85, its third annual conference, to be held in Edmonton, Alberta, March 14-16. A number of Logo presentations are scheduled, including one entitled "Physics and Math Simulations in Logo," by Tom Lough. For more conference information, write to Anne McIntyre, 5640 Ada Boulevard, Edmonton, Alberta T5W 4N8 Canada.

EduComp announces a series of summer Logo activities. If you want to learn how to explore Logo with your students, and to obtain practical ideas for scheduling, using teaching aids, creating bulletin boards, and establishing a Logo environment, read on. Beginning Logo is taught June 26-28, starts from scratch and progresses through variables. Continuing Logo, June 28-30, contains more variables, recursion, procedure refinement, conditionals, and beginning list processing. Both courses are 30 hours long, carry two credits, and cost $90 plus a $15 lab fee. The instructors are Christi Jaeger and Carolyn Green, authors of Teacher, Kids, and Logo. For more information, write: EduComp Summer Logo, 14242 Wyeth Avenue, Irvine, CA 92714.

J B Software has developed two interactive Logo programs for use with handicapped children. Logo Learning with Numbers and Logo Learning with Words are available for Logo versions on the Apple and Commodore 64 computers. For more information, write J B Software, 5813 Castano Drive, San Jose, CA 95129, or call (408) 996-9630.

The January issue of The Jeffries Report, an informative newsletter which reports on the state of health of various aspects of the microcomputer industry, indicated that the new Commodore 128 computer has an empty read-only memory (ROM) slot inside. Various rumors are circulating for how the slot may be used. Jeffries suggests that the chip eventually installed in the slot might turn out to be one for Logo! This means that, when the computer is turned on, Logo is ready to go. Let's wait and see on this one! (The Jeffries Report, Box 6838, Santa Barbara, CA 93160)

The January 28 issue of InfoWorld had a story on page 15 about the new generation of computers Atari just unveiled. Featuring a 68000 microprocessor, the ST line has been called a lower resolution color Macintosh at half the price. The article suggested that the machines will be sold with either BASIC or Logo! This may be a computer family that bears watching.

In the January 1985 issue of School Microcomputing Bulletin, an interesting lesson plan is presented for teaching students about artificial intelligence. Logo teachers who want to emphasize this aspect of the language might find the lesson plan useful. For more information, write Learning Publications, Inc., 303 Bay Drive North, Bradenton Beach, FL 33510.

Dan and Molly Watt write that Learning with Logo (Byte/McGraw-Hill) is now being translated into Spanish, German, Japanese, and Hebrew, and that their Teaching with Logo (Addison-Wesley) is scheduled for publication this spring. Also, Dan's Learning with Apple Logo is now available.

The Project of Equal Education Rights (PEER) announces the establishment of the National Center for Computer Equity to help ensure that computer opportunities are available to all students regardless of race, sex, or disability. For more information, call Leslie Wolfe at (202) 332-7337, or write to PEER, 1413 K Street NW, Washington, DC 20005.

EXSYS announces Turtlelab: Music, a pack of cards containing instructions and musical activities for Terrapin or Krell Logo. The eleven-card pack costs $7.95 plus $1.00 for shipping. EXSYS, 2728 23rd Street, Greeley, CO 80631. (303) 330-8021.

I have used Logo with my college physics classes for nearly two years. It gives me a wonderful feeling to experience the excitement in the lab when someone makes an interesting discovery and everyone else rushes over to see. When I hear my physics students planning their next Logo experiment or help them set up a procedure to simulate some physical situation they thought up, I get a sense that something important is happening to them.

I wish for all higher level students this same feeling. Now it looks as if there might be a chance they can catch it.

FD 100!

Tom

Logo Announcements

Harvard Associates has a number of exciting Logo announcements! First, their PC Logo is now available on the Sanyo 550 Personal Computer series. A minimum of 128K of memory and one disk drive is required. The cost is $149.95. Second, Harvard Associates has been selected to market the Valiant robot turtle, the first of a new breed. The Valiant has no cord connecting it to the computer! Instead, it uses an infrared communications link to pass computer commands. Thus, it can roam completely unencumbered, drawing complex designs or performing intricate choreography without your students having to worry about it getting tangled up in a cable. The Valiant is available for popular versions of Logo for the Apple II and IBM PC families, and the Commodore 64 computer. For more information, write: Harvard Associates, Inc., Attn: Valiant Turtle, 260 Beacon Street, Somerville, MA 02143, or call (617) 492-0660.

The Lisp Company offers versions of its TLC-Logo for several Z-80 based computers. Hailed as a second generation version, TLC-Logo possesses qualities of LISP and Smalltalk, yet is designed for a straightforward transition from the initial Logo versions in wide use today. TLC-Logo versions are available for the Epson QX-10, NCR Decision Mate 5, the NorthStar Advantage, and the Kaypro 284, 484, and 10. A Non-graphics customizable version is also marketed. For more information, write The Lisp Company, PO Box 487, Redwood Estates, CA 95044, or call (408) 354-3668.

Tipps for Teachers

by Steve Tipps

Survey Sense

The survey of surveys in last month's column suggested how Logo could be used as a data collection tool. For instance, information about birthdays or favorite pets might be collected with an interactive program with questions such as "Which month were you born?" One way to display the information in a kind of bar chart was also suggested.

Other ways of organizing, displaying and comparing information are also possible. Projects involving information in social studies, science, and mathematics from kindergarten through college level are easy to design. Tools which find high and low numbers, calculate the range, and compare results are useful. For tips on checking data as it is collected, see this month's Q and A column.

WHAT'S IN A LIST?

In the SURVEY procedure last month, information was collected as numbers for each month. You could check totals by printing out the names of the variables.

```
PO NAMES
"JAN IS 10
"FEB IS 8
....
"DEC IS 12
```

Or the value can be printed using PRINT and THING, a variable referent. In :JAN, the : is just short hand for the THING associated with the variable name JAN.

```
PRINT :JAN
10
PRINT THING "JAN
10
```

Working with the month names individually is not handy. A way to bind all the months together is needed. Here is an obvious place that a "list" of months is helpful.

```
MAKE "BIRTH.MONTHS [JAN FEB MAR APR MAY
JUN JUL AUG SEP OCT NOV DEC]
```

Tipps continued

Now you can ask questions about the list of months. What is the highest number of birthdays in any month? Which month has the most birthdays in this survey? How many responses where there altogether? What percent of birthdays were in each month? How does this survey compare to the survey last week? In each case, recursion and working with lists is used.

BIRTHDAYS MOST AND LEAST

Let's use a recursive procedure to make a chart about the number of birthdays in each month. The chart has five instructions, following the format shown in Tony Stavely's article on page 1 of this issue.

If the list is empty, stop.

Otherwise, print the name of the first month.

Print some spaces, or tab over.

Print the value attached to the first month.

Take the first month off the list and perform the same actions.

Notice how the CHART procedure follows this plan closely.

```
TO CHART :LIST
IF :LIST = [ ] STOP
PRINT1 FIRST :LIST
TAB 4
PRINT THING FIRST :LIST
CHART BUTFIRST :LIST
END

TO TAB :NUMBER
REPEAT :NUMBER [ PRINT1 CHAR 32]
END
```

You might want to remember which month had the high number. The recursive plan works, but the middle of the FIND.MOST procedure is different. For each month, the number of birthdays is compared with whatever value is presently the highest.

```
TO FIND.MOST :LIST
IF :LIST = [] RESULTS.MOST STOP
TEST THING FIRST :LIST > :MOST
IFTRUE MAKE "MOST THING FIRST :LIST
IFTRUE MAKE "MOST.MONTH FIRST :LIST
FIND.MOST BUTFIRST :LIST
END
```

```
     TO RESULTS.MOST
     PRINT (SE [THE MOST BIRTHDAYS WERE  IN]
:MOST.MONTH )
     PRINT (SE [THERE WERE] :MOST [BIRTHDAYS
IN] :MOST.MONTH )
     END
```

To start, you must first set the value of "MOST to 0 and the "MOST.MONTH to empty.

```
MAKE "MOST 0
MAKE "MOST.MONTH "
FIND.MOST "BIRTH.MONTHS
THE MOST BIRTHDAYS WERE IN APR
THERE WERE 13 BIRTHDAYS IN APR
```

The short variable name for the month winds up being a problem in the end.

Making a FIND.LEAST procedure is a simple variation. However, both procedures have not accounted for the possibility that two months might tie for high or low. Try writing a procedure for ties or pose it as a challenge for your students. Send me some of your solutions and ask for mine. Don't be hampered by the way I have organized the information. You may need to create other kinds of lists.

The high and low values are necessary to create another piece of information - the range. If the largest number of birthdays in a month was 13 and least number of birthdays was 6, the range would be 13 - 6 or 7.

```
TO RANGE
OUTPUT :MOST - :LEAST
END

PRINT RANGE
7
```

You also might write a procedure to correct the printed message in the "results" procedures if either :MOST or :LEAST equals 1.

You now have the beginning of a handy tool chest of procedures. The high, low, and range are very useful for describing many kinds of information.

CATEGORIES OF COUNTING

The information collected in the birthday survey is categorical data; the answers can be classified into groups. Although the months are in order, the order has no value associated with it. August is not a better month than July just because it comes after July. Even if we number the months 1 to 12, the order does not imply value. Other kinds

Tipps continued

of categorical data would be favorite names, pets, or ice cream. Each category is a collection of choices, not answers which have value such as grades of 70 or 90. Categories are analyzed by comparing the number of items in each category.

In a survey of ice cream favorites, six regular flavors and a seventh "other" category were used.

```
MAKE "FLAVORS [ VANILLA CHOCOLATE
STRAWBERRY PRALINE COCONUT MOCHA OTHER]
```

The results of the survey for Bowie School were charted and saved in a file.

```
CHART "FLAVORS
VANILLA    21
CHOCOLATE    14
STRAWBERRY    9
PRALINE    5
COCONUT    4
MOCHA    6
OTHER    21

ERASE PROCEDURES
SAVE "FLAVORS.AT.BOWIE
```

In Crockett School, the results were different and were saved in a different file.

```
CHART "FLAVORS
VANILLA    14
CHOCOLATE    7
STRAWBERRY    12
PRALINE    8
COCONUT    3
MOCHA    5
OTHER    11

ERASE PROCEDURES
SAVE "FLAVORS.AT.CROCKETT
```

One question might be whether the flavor favorites were similar or very different in the two schools. The numbers are certainly different, but that is partly because the number of students were different. Some way of comparing the results other than counting is needed.

PERHAPS PERCENTS

One way that categorical (or nominal) data is compared is with percentages. Percent is a statement of fractional, or proportional relationship. If you count 50 cars and 15 of them are red, then 15/50 is a fraction which expresses the relationship of reds to the total. Other ways to write the fraction is .30 or 3/10. The percentage is

Logo can calculate the percentages. To get the percentage, the total of all the responses is needed. A recursive procedure is also used for the TOTAL. OUTPUT is used in TOTAL because the results are going to be used in another procedure. Note the similarity of this procedure to Tony Stavely's ADDUP on page 1. It does the same thing as TOTAL.

```
TO TOTAL :LIST
IF :LIST = [] OUTPUT :SUM
MAKE "SUM :SUM + THING FIRST :LIST
OUTPUT TOTAL BUTFIRST :LIST
END
```

Now you can print totals. Read in one of the data files from the ice cream flavor survey and calculate the total. Then read the other file and calculate. Be sure to set "SUM to zero each time you want the total.

```
READ "FLAVORS.AT.BOWIE
MAKE "SUM 0
PRINT TOTAL :FLAVORS
80

READ "FLAVORS.AT.CROCKETT
MAKE "SUM 0
PRINT TOTAL :FLAVORS
60
```

Dividing the number in each category by the total gives the fraction.

```
MAKE "SUM 0
PRINT (THING FIRST :FLAVORS) / (TOTAL
:FLAVORS)
```

Multiplying by 100 changes the decimal fraction into a percent. Then percent can then be rounded to a whole number with the ROUND operation.

```
MAKE "SUM 0
PRINT ROUND (THING FIRST :FLAVORS) /
(TOTAL :FLAVORS) * 100
```

This is the one operational line in a recursive routine which prints out the percentages of each category. A new variable "ALL was initialized so that calculating TOTAL many times was eliminated.

```
TO SET.VALUES :LIST
MAKE "SUM 0
MAKE "ALL TOTAL :LIST
END
```

Tipps continued

```
    TO PERCENTAGES :LIST
    IF :LIST = [ ] STOP
    PRINT1 FIRST :LIST
    TAB 5
    PRINT1 ROUND (THING FIRST :LIST) / :ALL
* 100
    PERCENTAGES BUTFIRST :LIST
    END
```

Percentages may be more useful in comparing the results of a survey than are the actual numbers because they adjust for different total numbers of responses. Be certain that you know which data file you are using.

```
     READ "FLAVORS.AT.BOWIE
     SET.VALUES :FLAVORS
     PERCENTAGES :FLAVORS
     VANILLA    26
     CHOCOLATE     17
     STRAWBERRY     11
     PRALINE    6
     COCONUT     5
     MOCHA    7
     OTHER    26

     READ "FLAVORS.AT.CROCKETT
     SET.VALUES :FLAVORS
     PERCENTAGES :FLAVORS
     VANILLA    23
     CHOCOLATE     12
     STRAWBERRY     20
     PRALINE    13
     COCONUT     5
     MOCHA    8
     OTHER    18
```

The percentages make it possible to compare the results. About the same percent of students at both school liked vanilla, coconut, and mocha. Chocolate was a favorite at Bowie, while strawberry and praline made a good showing at Crockett. Several more schools can also be surveyed and the results compared by keeping separate data files. Although the comparisons are only interocular (or "eyeballed") the sense gained through these comparisons is very important in reading and understanding many different tables and charts in every day reading. Working through your own surveys in Logo helps with your survey sense.

Steve Tipps is the West Professor of Education at Midwestern State University in Wichita Falls, TX. He presents Logo workshops for school systems throughout the United States.

NLXionary

A Lectionary and Discussion of Logo Readings

by Griff Wigley

Time to liven things up, Logo lovers. If you haven't noticed, this column has been slowly changing over the last year. It started as sort of an annotated reader's guide to current and ancient Logo articles. As that became increasingly boring, I began editorializing more on each selection, first as to my judgement of its quality, and lately, using selections as vehicles for espousing my own perception of Logo reality.

Out of the goodness of my heart and the laziness of my bones, I want to afford NLX readers the same opportunity.

I am taking editor Lough's advice to stir the soup more. The column will become more interactive - sort of an electronic conference tree in print. I will sow the seeds of discussion by continuing to summarize current Logo articles. I will do my best to leave out my own editorializing until I have heard from those of you interested enough to comment on the articles. I especially encourage you to comment on the comments of others. Write to me at 918 College Street, Northfield, MN 55057. And now for this month's offerings.

"Logo Programming: Can It Change How Children Think?" by Douglas H. Clements, Electronic Learning, January 1985.

>Clements investigated the impact of Logo on children's cognitive abilities and cognitive development. Six pre- and post-tests were sandwiched around twelve weeks of Logo (two forty minute sessions per week) for a group of eighteen first graders, with a similar format for a control group of reading and math oriented computer assisted instruction (CAI).

>Results showed that, while there was no evidence of Logo affecting cognitive development any more than CAI, significant gains were indicated in the areas of problem solving abilities, creativity, and the abilities of the children to think about their own thinking, suggesting that Logo "may affect the way in which children use the cognitive abilities they possess."

NLXionary continued

"Logo Under Fire: A Conversation with Seymour Papert." by John Green, Classroom Computer Learning, January 1985.

>Just what the doctor ordered - an article touching on many of the controversies surrounding the educational use of computers, including Logo and artificial intelligence. Papert attacks the Bank Street study By Pea and Kurland, and yet is receptive to early results of a recent Canadian study by Higginson and Burnett, indicating a falling off of student competence with Logo in the second year. What might he say about the Clements study mentioned above?

"Is Computer Education Off Track? An Interview with Judah Schwartz." by Holly Brady and Melinda Levine, Classroom Computer Learning, February, 1985.

>Schwartz is co-director of the newly funded Educational Technology Center at Harvard. The Center's mission is to investigate ways in which various technologies can have a positive effect upon K - 12 math, science, and computer education. An advocate of the use of computers as tools, Schwartz is negative on teaching Logo in both elementary and high schools. He sees Logo as a "machine shop" in which any tool can be built, but that the work and skill to do so is unrealistic.

Griff Wigley is a co-owner of Family Computing, Inc., a facilitator in the Faribault (MN) Public Schools, and a school board member of Prairie Creek Community School.

TurtleTips

by

Donna Lanyi
and Jane Toth

All of the "turtling around" during the school year certainly has produced some interesting graphic pictures. However, these terrific ideas are lost when the computer is turned off unless they are either saved as pictures or written and saved as procedures.

We have found that introducing the concept of procedure writing is fairly straightforward. The concept is capable of being handled by primary-age students. The difficulty comes when teaching the concept of editing. And, of course, editing is an integral part of procedure writing. Here are some thoughts for helping your students become Logo editors.

TEACHER HINTS

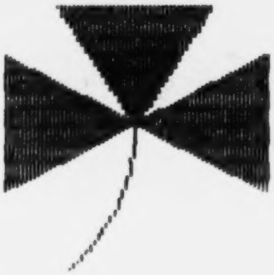
Students seem to be able to master editing fairly easily if only a few editing commands are presented at one time. Introduce only the necessary editing commands at first, and allow students to become familiar with the editing concept. As student editing skills are developed, introduce the more advanced editing commands and capabilities.

We post a list of these essential editing commands by each computer. Our students also have a personal list in their computer notebook. We have used a teacher prepared worksheet containing procedures with deliberate errors. Students are asked to define the procedures exactly as presented, including the errors, and then to use the editor to make corrections. (See also "Brookline Students Hunt Logo Bugs," by Jay Sugarman, NLX, December 1982.)

PROCEDURE OF THE MONTH

May you find your pot of gold at the end of the rainbow! For a St. Patrick's Day celebration, let your students design (and edit) procedures with an Irish theme - leprechauns, shamrocks, pots of gold, and rainbows. Our shamrock procedure can be used as a starting point for a St. Patrick's Day mural or bulletin board display.

Here is also an opportunity to make use of some of the modularity ideas you have already studied. For example, break the shamrock into parts and write a procedure for each part.



If you are interested in a listing of this procedure, please send us a self addressed stamped envelope. We invite any elementary teachers or students to send their procedures or ideas to be shared. Donna Lanyi and Jane Toth, Beall Avenue School, 716 Beall Avenue, Wooster, OH 44691.

Donna Lanyi and Jane Toth use Logo in their elementary classrooms in the Wooster (OH) City Schools.

Listful Thinking

by

Glen Bull and Paula Cochran

Powerful Selections

Last month we introduced the idea of selecting an item from a list. SELECT, you will remember, is a procedure that picks an item from a list, and then deletes the item. To use SELECT, first we have to make a list.

```
?MAKE "TOYS [BALL BEAR DRUM WAGON]
?PRINT :TOYS
BALL BEAR DRUM WAGON
```

Now we can SELECT a toy.

```
?PRINT SELECT "TOYS
DRUM
```

After the toy is selected, it is no longer in the list.

```
?PRINT :TOYS
BALL BEAR WAGON
```

Since SELECT deletes each choice from the list after selecting it, the same item is never chosen twice. This eventually destroys the list. For that reason, last month we also gave you a procedure for creating a list from the text of a procedure. The text is simply entered into a procedure, without PRINT or brackets.

```
TO ANIMAL.TEXT
RABBIT
FOX
DEER
SQUIRREL
POSSUM
CHIPMUNK
SKUNK
END
```

Then the CREATE procedure creates a list from the text.

```
    ?CREATE "ANIMAL.LIST "ANIMAL.TEXT
    ?PRINT :ANIMAL.LIST
    [RABBIT]  [FOX]  [DEER]  [SQUIRREL]
[POSSUM] [CHIPMUNK] [SKUNK]
```

There are two advantages to creating a list from text entered in the form of a Logo procedure (such as ANIMAL.TEXT):

(1) Large amounts of text are easier to enter using the Logo editor.

(2) It is easy to use CREATE to create the list a second time after all the items in the list are selected.

To see how this works, let's use SELECT with ANIMAL.LIST.

```
    ?PRINT SELECT "ANIMAL.LIST
    DEER
    ?PRINT :ANIMAL.LIST
    [RABBIT]  [FOX] [SQUIRREL]  [POSSUM]
[CHIPMUNK] [SKUNK]
```

DEER has been SELECTed and removed from the list. Let's try it again.

```
    ?PRINT SELECT "ANIMAL.LIST
    CHIPMUNK
    ?PRINT :ANIMAL.LIST
    [RABBIT]  [FOX] [SQUIRREL]  [POSSUM]
[SKUNK]
```

This time CHIPMUNK has been SELECTed and removed from the list. Eventually, SELECT will select and remove all the animals from the list. The list will remain empty until you use CREATE to create the list again for another activity.

Items in your list can consist of more than one word. Here is a text procedure that contains phrases.

```
TO PLACE.TEXT
BY THE POND
BETWEEN THE BUSHES
UNDER THE TREE
ON PINE NEEDLES
BENEATH WHITE CLOUDS
IN THE WATER
IN THE GRASS
END
```

Use CREATE to make a list of places.

```
    ?CREATE "PLACE.LIST "PLACE.TEXT
    ?PRINT :PLACE.LIST
    [BY THE POND] [BETWEEN THE BUSHES]
[UNDER THE TREE] [ON PINE NEEDLES] [BENEATH
WHITE CLOUDS] [IN THE WATER] [IN THE GRASS]
```

Now SELECT can be used to select a place.

Listful Thinking continued

```
?PRINT SELECT "PLACE.LIST
ON PINE NEEDLES
```

The same strategy can be used to create a list of adjectives.

```
TO ADJECTIVE.TEXT
STILL
QUIET
NOISY
ALERT
QUICK
SILENT
YOUNG
END
```

```
?CREATE "ADJECTIVE.LIST "ADJECTIVE.TEXT
?PRINT :ADJECTIVE.LIST
[STILL] [QUIET] [NOISY] [ALERT] [QUICK]
[SILENT] [YOUNG]
```

THE VERSE THINGS IN LIFE ARE FREE

What can you do with lists using SELECT? One thing you can do with them is create free verse. With SELECT each verse will be different. Here is a short procedure that creates free verse about the woods.

```
TO WOODS.POEM
PRINT SELECT "ANIMAL.LIST
PRINT SELECT "ADJECTIVE.LIST
PRINT SELECT "PLACE.LIST
END
```

A verse is created each time you type WOODS.POEM.

```
?WOODS.POEM
DEER
QUIET
UNDER WHITE CLOUDS

?REPEAT 2 [ WOODS.POEM PRINT [] ]
SQUIRREL
QUICK
ON PINE NEEDLES

FOX
ALERT
IN THE GRASS
```

Eventually, all the elements in the lists will be used up. It simplifies matters to make a master procedure that creates all the lists again.

```
TO CREATE.LISTS
CREATE "ANIMAL.LIST "ANIMAL.TEXT
CREATE "PLACE.LIST "PLACE.TEXT
CREATE "ADJECTIVE.LIST "ADJECTIVE.TEXT
END

?CREATE.LISTS
```

The CREATE.LISTS procedure can be used by itself, or it can be placed in a procedure that writes verses.

```
TO FREE.VERSE
CREATE.LISTS
REPEAT 7 [ WOODS.POEM PRINT [] ]
END
```

This free verse program writes poems about the woods. Other free verse programs could write poems about the circus or the zoo, with different kinds of animals. Or, if the class is studying amphibians in science class, students could create poems about alligators and lizards and salamanders in English class.

ONCE UPON A TIME...

You can use the concept of SELECT to explore the idea of story sequencing. First create some text that conveys a story. This text is a story about an encounter with a mean dog. When you enter the text, be sure to press the RETURN key at the end of each line.

```
TO ENCOUNTER.TEXT
SALLY SAW THE MEAN DOG COMING.
SHE TURNED AND STARTED RUNNING AWAY.
THE DOG CHASED HER DOWN THE STREET.
END
```

After the text has been defined, it is necessary to use CREATE to create a list from the text before using it with SELECT.

```
?CREATE "ENCOUNTER.LIST "ENCOUNTER.TEXT
?PRINT SELECT "ENCOUNTER.LIST
SHE TURNED AND STARTED RUNNING AWAY.

?PRINT SELECT "ENCOUNTER.LIST
SALLY SAW THE MEAN DOG COMING.
```

In this instance, the events in the story are out of sequence. This is shown more clearly in a procedure that produces the entire story.

Listful Thinking continued

```
TO ENCOUNTER.STORY
CREATE "ENCOUNTER.LIST "ENCOUNTER.TEXT
REPEAT 3 [PRINT SELECT "ENCOUNTER.LIST]
PRINT []
END
```

Here are some possible ENCOUNTER stories. The first story is clearly out of sequence. Sally starts running away before she sees the dog.

```
?ENCOUNTER.STORY
SHE TURNED AND STARTED RUNNING AWAY.
THE DOG CHASED HER DOWN THE STREET.
SALLY SAW THE MEAN DOG COMING.
```

The next story is more plausible, but still does not read as smoothly as it might.

```
?ENCOUNTER.STORY
SALLY SAW THE MEAN DOG COMING.
THE DOG CHASED HER DOWN THE STREET.
SHE TURNED AND STARTED RUNNING AWAY.
```

The final story is in the best order.

```
?ENCOUNTER.STORY
SALLY SAW THE MEAN DOG COMING.
SHE TURNED AND STARTED RUNNING AWAY.
THE DOG CHASED HER DOWN THE STREET.
```

This activity can be used as a springboard for discussing the importance of placing written information in the proper sequence. Stories the children themselves have written could easily be explored this way.

Another way to use SELECT is to create story starters. Of course, you don't have to limit yourself to just three sentences. You could create a list of a dozen sentences, and then SELECT three of them for each story starter. Or you could store different stories under different procedure names.

POWER TOOLS DISK OFFER

We have been placing various Logo versions of key procedures end of our column. For example, SELECT and CREATE were introduced last month, and are also included at the end of this column. (A slight improvement has been made in SELECT.) However, we recognize that some readers may not have the time to type in the procedures they need. Therefore, on a trial basis, we would like to offer this month's procedures on a disk. Send us a formatted blank disk and stamps or international coupons for return postage. This offer expires March 31, 1985.

NLX March Story Tools
Curry School of Education
University of Virginia
Ruffner Hall, 405 Emmet Street
Charlottesville, VA 22903

When you send your blank disk, be sure to indicate which version of the procedures you want:

(1) IBM Logo
(2) Apple Logo
(3) Apple Logo II
(4) Terrapin 1.0 versions
(5) Terrapin 2.0
(6) Krell Logo
(7) Commodore Logo

PROCEDURES

PICK and CREATE are the same for all Logo versions listed above.

```
TO PICK :LIST
OP ITEM 1 + (RANDOM COUNT :LIST) :LIST
END

TO CREATE :LIST.NAME :PROCEDURE
MAKE :LIST.NAME BF TEXT :PROCEDURE
END
```

Here are the remaining procedures for IBM Logo and Apple Logo.

```
TO SELECT :LIST
IF THING :LIST = [] [OUTPUT [] ]
MAKE "ITEM PICK THING :LIST
DELETE :ITEM :LIST
OUTPUT :ITEM
END

TO DELETE :ITEM :LIST
MAKE  :LIST DELETE.LOOP :ITEM THING :LIST []
END

TO DELETE.LOOP :ITEM :OLDLIST :NEWLIST
TEST EMPTYP :OLDLIST
IFTRUE [PRINT SE :ITEM [NOT FOUND]]
IFTRUE [OUTPUT :NEWLIST]
TEST :ITEM = FIRST :OLDLIST
IFTRUE [OP SE :NEWLIST BF :OLDLIST]
OP DELETE.LOOP :ITEM BF :OLDLIST LPUT  FIRST
    :OLDLIST :NEWLIST
END
```

Listful Thinking continued

Here are the remaining procedures for Terrapin Logo on the Apple and Commodore 64 computers.

```
TO SELECT :LIST
IF THING :LIST = [] OUTPUT []
MAKE "ITEM PICK THING :LIST
DELETE :ITEM :LIST
OUTPUT :ITEM
END

TO DELETE :ITEM :LIST
MAKE :LIST DELETE.LOOP :ITEM THING :LIST [ ]
END

TO DELETE.LOOP :ITEM :OLDLIST :NEWLIST
TEST :OLDLIST = []
IFTRUE PRINT SE :ITEM [NOT FOUND]
IFTRUE OUTPUT :NEWLIST
TEST :ITEM = FIRST :OLDLIST
IFTRUE OP SE :NEWLIST BF :OLDLIST
OP  DELETE.LOOP :ITEM BF :OLDLIST LPUT FIRST
     :OLDLIST :NEWLIST
END
```

In Terrapin Version 1, PICK also requires these procedures:

```
TO ITEM :NUMBER :LIST
OUTPUT ITEM.LOOP :NUMBER :LIST 1
END

TO ITEM.LOOP :NUMBER :LIST :COUNT
IF :LIST = [] OP (SE [THERE ARE ONLY] :COUNT
     [ITEMS IN THIS LIST.] )
IF :NUMBER = :COUNT OP FIRST :LIST
OP  ITEM.LOOP  :NUMBER BF :LIST :COUNT  +  1
END

TO COUNT :LIST
OUTPUT COUNT.LOOP :LIST 0
END

TO COUNT.LOOP :LIST :NUMBER
IF :LIST = [] OUTPUT :NUMBER
OP COUNT.LOOP BF :LIST :NUMBER + 1
END
```

Glen Bull is a professor in the University of Virginia's School of Education, and teaches Logo courses at both the graduate and undergraduate level. Paula Cochran is a reformed English major who studied linguistics at the University of Cambridge. She is now a speech-language pathologist working with language-disabled children.

Logo Disserts:

Dissertations Dealing with Logo

by Barbara Elias

Can an inservice training program serve as an effective means for preparing teachers to use Logo as well as assist them in designing and using a Logo curriculum? This question was investigated in an exploratory study involving eleven elementary and middle school teachers. Previous microcomputer experience ranged from little or none to a maximum of two computer education courses.

A one week intensive workshop on the Logo language itself and how to teach a Logo centered curriculum was used for the first phase of teacher training. Workshop sessions included experiences with modular programming, iteration, variability, and planning.

The implementation phase included follow-up weekly observation and feedback conferences ("coaching") with each teacher. This phase also incorporated teacher learning team sessions for sharing experiences and providing additional support. A variety of materials were provided for the teachers.

Teacher interviews, self report records, and Logo skills test scores provided data for the study. Results indicated that most teachers developed a measure of expertise with Logo and success with the Logo curriculum. The results also lend support to the view that teachers who are involved in introducing Logo and Logo curriculum materials need training as well as follow-up support.

>>>Ferres, G. W. (1984). Training and Implementation Strategies Appropriate to the Introduction of Logo into Teachers' Curriculum and Instruction. (Doctoral dissertation, University of Oregon, 1983). Dissertation Abstracts International, 44, 3264-A.

(Ed. Note: Each month, Barbara Elias highlights a dissertation or thesis dealing with Logo. She is accepting copies of recent research results for the NLX Dissertation and Thesis Repository. Reports may be mailed to her in care of NLX, PO Box 5341, Charlottesville, VA 22905.)

Barbara Elias is an assistant professor in the Education Department of Virginia State University in Petersburg, VA, and a doctoral candidate at the University of Virginia.

Teacher to Teacher

Teacher Training Materials with Classroom Applications

reviewed by
Anne Cairns Federlein

Resolutions! At midnight, December 31, 1984, the co-editors of this column resolved, among other things, to continue the quest to find exciting Logo classroom materials and teachers who are using them. This is the only resolution which my three colleagues have allowed me to divulge publicly. But I can tell you this ... it is the only resolution they have kept!

ANOTHER MECC FIND!

Teachers new to Logo will find Introduction to Logo for Teachers Training Materials by Marcia Horn of the Minnesota Educational Computing Corporation (MECC) an answer to their innermost fears! Although designed to be used in a Logo teacher training course, the book is a most worthwhile addition to any Logo classroom as well. Let me tell you why.

The primary emphasis of the book is to help teachers emphasize the problem solving process with Logo, not necessarily concentrating on the language itself.

Modules for Apple Logo divide the book, beginning with "Meeting the Turtle," and "Discovering Patterns," then progressing to "Teaching Turtles," "Using the Editor," "Changing Pen and Background Colors," and "Creating Building Blocks." More difficult lessons include "Procedures," "Recursion," and "Words and Lists."

The transparency masters in Section II enrich the lessons with good visual aids, and are accompanied by suggestions on how to develop a Logo journal, reinforce new vocabulary, and guide groups on working effectively.

The activity sheet section contains projects that are as open ended as we have reviewed to date. For example, one sheet suggests a format for a Logo journal, with statements such as "What I want to do," "What I did," and "What I learned," giving guided structure and valuing the child's responses, yet avoiding rote "right" answers. Another encourages a team of children to record the steps in their thinking as they design procedures.

HELP! HELP!

The reference section is priceless to the teacher struggling with Logo and computer operations. Apple Logo short word cards, editor cards, a Logo command summary, and diskettes for both students and teacher courses fill immediate needs. Long term needs are addressed by references to books, other classroom materials, magazines, user groups, newsletters, and the MECC hotline number.

This book is highly recommended for the beginning teacher in Logo. I wish it could be included in every educational sale of an Apple II computer!

Introduction to Logo for Teachers Teaching Materials is available for $49 from MECC, 3490 Lexington Avenue North, St. Paul, MN 55112, or call (612) 481-3527.

We welcome reader comments and opinions on this review. Send your comments to Logo Information For Teachers (LIFT), PO Box 5396, Plymouth, MI 48170.

Anne Cairns Federlein is a professor in the Early Childhood Area, School of Human and Educational Services, Oakland University, Rochester, MI.

Logo Notes

Logo Computer Systems Incorporated (LCSI) announces a price reduction of $100 for its Sprite Logo for the Apple. The new price is now $199 per copy, and includes a plug-in Sprite board, language disk, and three manuals. Quantity discounts are available. For more information, write LCSI, 555 West 57th Street, Suite 1236, New York, NY 10019, or call (212) 765-4780.

Terrapin, Inc., has available a networked version of its popular Logo version for the Apple II personal computer family. Configured for the Corvus hard disk system, which allows up to 64 computers to be interconnected, the networked Logo provides a flexibility not available to teachers using individual computers. For more information, write Terrapin, Inc., 222 Third Street, Cambridge, MA 02142, or call (617) 492-8816.

K-12 Micromedia offers a free 24-page catalog of Logo books, teaching aids, and utilities. This is the only Logo collection of its kind, and contains Logo goodies for the novice and expert alike. To order a copy, write to K-12 Micromedia, 172 Broadway, Woodcliff Lake, NJ 07675.

前 100

by
Hillel
Weintraub

Many Japanese companies recognize the potential of the educational market for software and computer products. Drill and practice materials are in demand by the "juku" (tutoring schools where students study after regular school hours.). Several companies are working to develop computer materials for these numerous private schools.

Personally, I have great doubts about the overall effectiveness of this type of education. It seems to be an example of the dissociated learning to which Papert referred in Mindstorms. Yet, the Japanese children seem to develop a certain kind of powerful involvement due to the competitive atmosphere of these schools and of the educational system in general. Winning this memorize-it-spit-it-out-on-the-test game is more important than learning to most students. (Ed. Note: For more information, see "Japanese Education and Its Implications for US Education," by Nobuo K. Shimahara, Kappan, February 1985, and "Of Sushi and Mushi," by Duane Yee, NLX, January 1984))

ONE COMPANY STEPS OUT

I'd like to tell you about one innovative company which has taken a rather daring step into educational areas. Three years ago, Uny Company (one of Japan's largest retail organizations) formed its BYNAS division, devoted to developing and providing application systems, and acting as a liaison between users, scholars, educators, and manufacturers. As part of their educational computing interests, they have become a leader in Logo work in Japan. But BYNAS was not established with the expectation of immediate results.

Beginning as the Japanese representative for Terrapin Logo, BYNAS is now hoping to represent all Logo versions equally, including any they develop. This seems to put them in a rather unusual position of representing competing products. However, their intent is the growth of Logo-like education rather than the growth of a particular brand of Logo. More and more companies are showing a willingness to cooperate with them as this intention becomes more clear.

LOGO JOURNAL

An especially interesting BYNAS development is the publication of Logo Journal, which began last February. This Japanese language journal is making a serious effort to share information on all Logo versions as well as keep its readers up to date on new educational and commercial Logo happenings. But it is difficult for a company publication to gain the status and respect normally reserved for a university scholarly journal or other independent publication.

As a contributing editor, I write occasional articles for them or help with some English expressions in translations into Japanese.

The BYNAS staff members keep abreast of foreign published Logo materials and Logo developments in Japan. Agreements with both the NLX and YPLA newsletters and The Computing Teacher magazine allow for the translation and reprinting of articles of interest to the Japanese Logo community.

All articles are in Japanese, but most procedures are listed in English. With a circulation of about 350, this 16-page publication is available by subscription for $50 per year, air mail postpaid. For more information, write Uny Co., BYNAS Division, 3rd Floor, Dainagoya Bldg., 28-12 3-Chome, Nakamura-ku, Nagoya, Japan 450.

Until next time, MAE 100!

Hillel Weintraub is a teacher at Doshisha International High School in Kyoto-fu, Japan, and is the president of the Society for Microcomputing in Life and Education (SMILE).

Logo Notes

Are you a member of a Logo user group? If so, make sure your group is included in the first edition of The National Logo Exchange's Logo User Group Directory. Scheduled for publication this summer, the directory is planned to help establish communications between user groups, and to provide information to prospective members. A courtesy copy of the directory is sent to each participating user group. Send the name and address of your group, the name and telephone number of a contact person, the size of your membership, your objectives and interests, and a schedule of your meetings to NLX User Group Directory, PO Box 5341, Charlottesville, VA 22905.

Q and A

by
Jim McCauley

Q: I've been trying to learn how to trap errors in user input in Logo programs. For example, with this piece of Logo code,

```
     PRINT [GIVE ME A NUMBER AND I'LL ADD 20
TO IT.]
     MAKE "NUM FIRST READLIST
     PRINT :NUM + 20
```

If someone types in something other than a number (like the word FRED, say), the program dies because "+ DOESN'T LIKE FRED AS INPUT" or some such reason. I know how to do this in BASIC using loops and IF statements, but doing it in Apple Logo seems to involve the use of the commands CATCH and THROW, which I find really hard to figure out. Can you help?

<[-?-]>

A. Yes, I'll try. But let me suggest an approach other than CATCH and THROW, which are very complicated indeed and which do not appear in all Logo dialects.

The idea of error-trapping is to make sure that incorrect types and forms of data do not sneak into programs and bomb them in the way you describe. In languages like BASIC, this information enters the program through something like an INPUT statement.

```
     100 PRINT "GIVE ME A NUMBER AND I'LL
ADD 20 TO IT."
     110 INPUT N
     120 PRINT N + 20
```

Note the similarities to your Logo code fragment. If this BASIC program is run and the user types in something besides a number (such as FRED) instead of a number, the program will choke. To make sure this doesn't happen involves checking the type of input before it is passed on to other parts of the program.

To guard against repeated errors in input, a checking routine is usually placed in a loop from which there is no escape until the type and form of the input are correct.

While it is possible to create similar checking loops in Logo, there is a very different way of thinking about this kind of problem. Consider this: Logo programs can be thought of as "strong" when they "take care of themselves" with internal methods of guarding against errors commited by users. In Logo, we program in terms of procedures. How can we create procedures that are "strong?"

LOGO FILTER

One way would be to develop a "functional filter," an operation that captures incoming data and inspects it. Data of the correct type and form are allowed to pass through, while erroneous input is trapped, requiring the user to reenter the data. In Logo, the trick to doing this in elegant fashion is to think backwards!

Look at it this way: READLIST (or REQUEST) is an operation; it outputs a list of what the user types in. Here is a simple graphic of this action:

```
<-- READLIST
```

The arrow indicates the direction of the output, back toward the command or operation that called READLIST.

Let's take this Logo classic

```
TO BOX :SIZE
REPEAT 4 [FD :SIZE RT 90]
END
```

and use it to make an interactive box drawing program.

```
TO INTERBOX
PRINT [HOW BIG A BOX DO YOU WANT?]
BOX FIRST READLIST
END
```

The last line of the program executes like this. BOX waits for a number from FIRST, which waits for a number inside a list from READLIST, which waits for a number from the user. As long as the user types in a number, all is well. The number gets passed back to BOX eventually, like this.

```
BOX <-- FIRST <-- READLIST
```

However, if the user types in something other than a number, it gets passed into BOX where, within BOX, FORWARD protests that :SIZE isn't carrying a number.

Q and A continued

LOGO RECOGNIZERS

Logo has some built in "recognizers" that can inspect data objects and identify them. One of these is NUMBERP (or NUMBER?). It returns TRUE if its input is a number and FALSE if it receives anything else. We can use this primitive as a tool to trap erroneous input to INTERBOX when it runs READLIST if we arrange for a checking operation containing NUMBERP to inspect the data typed in before it can get inside the BOX procedure.

Remember that the data travels backwards from READLIST to FIRST to BOX. Suppose we stick in a checking procedure called CHECK.NUM in this chain of operations.

```
BOX <-- CHECK.NUM <-- FIRST <-- READLIST
```

CHECK.NUM intercepts FIRST's output before it can get into BOX, thus trapping bad data and allowing the good stuff to go through.

CHECK.NUM is going to take one input (a number, we hope!) and test it with NUMBERP. If the input passes the test, it is simply output to whatever calls CHECK.NUM (in this case, BOX).

```
TO CHECK.NUM :OBJ
IF NUMBERP :OBJ [OUTPUT :OBJ]
...
```

So far, so good. What should CHECK.NUM do if the input is not a number? Well, the user should be asked to type in good data.

```
PRINT [PLEASE ENTER A NUMBER.]
```

Now we want the user to type in a number, so we need another READLIST in here somewhere. But suppose that our user once again makes a mistake and types in a letter instead of a number? Oh, dear! We need to check that input, too!

Feels recursive, yes? Through the clever use of recursion, we can complete CHECK.NUM in one more line.

```
OUTPUT CHECK.NUM FIRST READLIST
```

Oho! CHECK.NUM uses another copy of itself to check its own input! Makes one dizzy, this recursive stuff...

Bombproof programs can be written in Logo. By combining Logo's other recognizer operations (such as LISTP or LIST?) with the Boolean operations (AND and OR, or ALLOF and ANYOF), it is possible to filter any kind of user input.

Here's a challenge for you. Write a filter procedure called CHECK.RANGE which accepts three inputs: a number to be checked, a low number, and a high number. CHECK.RANGE should allow only numbers between the high and low value inclusive to pass through. If other input is received, it prompts the user to reenter the data.

(Ed. Note: If you have a question about Logo programming techniques or the thinking which leads to Logo programming, send it to NLX Q&A, PO Box 5341, Charlottesville, VA 22905. Each month, Jim McCauley answers selected questions in this column.)

Jim McCauley is a Coordinator of Computer Education for the Santa Clara (CA) County Office of Education, and has written Logo articles for many national publications.

NLXual Challenges

by
Robs Muir

What? If?

One of the oft repeated advantages of Logo over other educational computing languages is its extensibility. In this context, extensibility refers to the ability to expand Logo's "vocabulary" beyond its basic vocabulary by defining procedures. The name of each new procedure is a word which extends Logo's expressibility. Contrast Logo with what John Allen calls "general purpose" languages, such as BASIC.

Logo's original foundation rests on a core of primitives (or basic vocabulary) that forms the environment in which users write their procedures. As the language has matured, we are beginning to see versions of Logo with expanded sets of primitives to lower Logo's threshold, or ease of use. Most versions of Logo now include "primitives" such as ITEM, COUNT, and DOT, which could be written as procedures reasonably easily. Indeed, most Logo tutorials recommend learning how to write versions of these primitives as a good way to learn how the language works.

NLXual Challenges continued

WHAT'S THE QUESTION?

The question of importance here is, "Are there some primitives that you just can't do without?" What about IF? Can you write an IF procedure in Logo, or is IF a "primal" primitive?

Michael Tempel, as Director of Training for Logo Computer Systems Incorporated (LCSI), often presents the challenge of writing Logo primitives in Logo in the advanced workshops he teaches. After kicking around the IF question at the LCSI offices in new York and Montreal for several months, he and others have composed a number of ways of writing an IF procedure.

In the April issue of the NLX, Michael Tempel and Mario Bourgoin of LCSI will share the first few IF's which they and other LCSI staffers created. In the May issue, they plan to present the rest, along with a brief discussion of how Logo makes decisions. However, they would also like to see what our readers can do. Let's not disappoint them!

THE IF RULES

Here are some rules and guidelines in writing your IF procedure.

-No fair using TEST, IFTRUE (IFT), or IFFALSE (IFF), since they make up another form of IF.

-Write a two-input version (if-then) for starters.

```
TO IF.THEN :CONDITION :TRUE.LIST
```

-Then tackle the three-input version (if-then-else).

```
TO IF.THEN.ELSE :CONDITION :TRUE.LIST
:FALSE.LIST
```

-This challenge doesn't officially contain an invitation for an IF which takes an optional third input. This would be quite difficult, since user-defined procedures require a definite number of inputs. You might think about the IF THEN ELSE form of MIT Logo (an older ALGOL patterned syntax), but that also would probably be very hard to write. Why do you think these are called NLXual Challenges?!

```
IF YOU.HAVE.A.SOLUTION [MAIL :SOLUTION]
[KEEP.TRYING!]

TO MAIL :SOLUTION
SEND.TO
MICHAEL.TEMPEL
LCSI
555.WEST.57TH.STREET
SUITE.1236
NEW.YORK.NY.10019
END

TO KEEP.TRYING!
CO
END
```

Robs Muir is a teacher in the Claremont CA Unified School District and Claremont Graduate School.

Special Education Logo Users Group Announced

Barbara Luetke-Stahlman announces the formation of a Special Educators Logo Users Group. The objectives include setting up networks, sharing research results, and formulating grant proposals for Logo projects in special education. For more information, write to her at the Hearing Impairment Program, Northern Illinois University, DeKalb, IL 60115.

The National Logo Exchange, copyright (c) 1985 by Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly, September through May, $25 (US) per year, mailed first class from Charlottesville, VA. $5 (US) additional per year for addresses outside the USA, Canada, and Mexico. The opinions expressed by the authors are not necessarily those of The National Logo Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress Street, Salem, MA 01971. Address other correspondence to: The National Logo Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor..Tom Lough

THE NATIONAL
LOGO
EXCHANGE

Volume 3 Number 8
FORWARD 100!
April 1985

The Pause That Debugs

by
Harold Nelson

It happened during a recent seminar discussion. We were examining various methods of debugging Logo programs. Some users felt that STEP and TRACE are absolutely necessary tools for debugging and should be included in all Logo versions. But, as the discussion developed, it became clear that we had failed to promote the proper understanding and use of PAUSE as the dynamic, interactive debugging tool that it is.

STEP, TRACE, and PAUSE

STEP runs a line of a program and then waits for you to press a key before running the next line. It is sort of like watching a football game on a special video tape machine that stops after each play, allowing you to study it; when you press a button, the next play is run.

TRACE shows you an explanation of the result of each line of a program as it is being run. It is like watching the football game on the video machine accompanied by a good commentary, telling you what caused the results you saw.

Both of these are good debugging tools, but they do not allow you to interact with the program while it is running. When you can do that, powerful debugging is possible. Your students should experience this power.

PAUSE facilitates such debugging. Let's take a closer look.

CAUSING PAUSING

There are two ways to cause a pause while a Logo program is running: press the appropriate key(s), such as Control-Z for Logos on the Apple computer, or use PAUSE inside a procedure. Here is an example which uses both ways.

The COUNTDOWN procedure counts down from an input number, but it has a bug: it never stops.

```
TO COUNTDOWN :COUNT
PRINT :COUNT
COUNTDOWN :COUNT - 1
END
```

```
COUNTDOWN 5
5
4
3
2
1
0
-1
-2
...
```

Press Control-Z (or whatever) to make the procedure pause. Logo prints a message telling you where it is pausing, and waits for an instruction.

```
PAUSING... IN COUNTDOWN:
PRINT :COUNT
COUNTDOWN ?
```

Now it is possible to do a number of interesting things not possible with STEP or TRACE which can help with debugging. For example, you could verify the values of any active global or local variables, such as that of :COUNT.

```
COUNTDOWN ? PONS
COUNT IS -21 (or some other value)
COUNTDOWN ?
```

Now let's edit COUNTDOWN while it is pausing, and add PAUSE to it!

```
COUNTDOWN ? EDIT "COUNTDOWN
```

Change the procedure until it looks like the one on the following page, and leave the editor in the usual way. Note that you are still in the paused condition after the COUNTDOWN procedure is defined.

Pause continued

```
TO COUNTDOWN :COUNT
PAUSE           <--This line added
PRINT :COUNT
COUNTDOWN :COUNT - 1
END

COUNTDOWN DEFINED
COUNTDOWN ?
```

Next, let's reset the value of COUNT to 3, and then resume running the procedure (with CO), in order to catch the bug.

```
COUNTDOWN ? MAKE "COUNT 3
COUNTDOWN ? CO
PAUSING... IN COUNTDOWN:
PAUSE
COUNTDOWN ?
```

By entering CO after each COUNTDOWN ?, you can step through the program. Now, however, when the count goes through zero, you can identify the bug, edit the procedure once more, and replace the PAUSE with something like IF :COUNT < 0 [STOP].

This rather obvious bug serves as a simple example to show how PAUSE can be used. Think about what you have done. You have just edited and run a procedure inside of a pause of that very procedure in order to debug it! I think that this is really amazing.

TURTLE PAUSE

Now suppose you were improving a recursive interactive turtle graphics program, such as the following.

```
TO DRAWIT
SETHEADING TOWARD READCHAR
FORWARD 5
DRAWIT
END

TO TOWARD :DIRECTION
IF :DIRECTION = "U [OUTPUT 0]
IF :DIRECTION = "R [OUTPUT 90]
IF :DIRECTION = "D [OUTPUT 180]
IF :DIRECTION = "L [OUTPUT 270]
OUTPUT TOWARD READCHAR
END
```

Type it in and work with it a while. Then make it pause from the keyboard and move the turtle home.

```
PAUSING... IN DRAWIT:
SETHEADING TOWARD READCHAR
DRAWIT ?PENUP HOME PENDOWN
DRAWIT ?
```

This demonstrates that you can change the turtle's state without stopping and rerunning the program.

Now type CO. The program crashes and Logo prints a complaint. This is because this particular program was expecting a character when it continued. Here is how to take care of that. Type DRAWIT and then make it pause once more.

```
PAUSING... IN DRAWIT:
SETHEADING TOWARD READCHAR
DRAWIT ?
```

Now type CO and an input, such as "R. The program continues and the turtle immediately executes a move to the right.

```
DRAWIT ? CO "R
```

Now you may decide to add a key H to make the turtle return to the home position. Make the program pause from the keyboard and type EDIT "TOWARD. Add this line to the procedure and redefine it.

```
IF :DIRECTION = "H [PENUP HOME PENDOWN]
```

When you leave the editor, you may note that you have lost your original picture and that Logo has delivered a mild complaint, depending on your particular version of Logo. But the purpose of this example is to demonstrate the interactive debugging power which is available to you the next time you are trying to fix up a more complex graphics program.

CONCLUDING THE PAUSE

In our seminar debugging discussion, we came to think of PAUSE as a sort of "time out" -- a special kind of interlude in which you can not only analyze the game so far, but during which you can also interact with and change the game. You can examine the values of variables, even local variables (inputs to procedures), assign and test new values, and inspect and edit the procedures being used.

Such powerful interactive debugging is well worth investing a little time to learn it and to teach it.

Harold Nelson is currently completing a book on advanced Logo programming concepts. He conducts Logo workshops around the world.

April 1985

NLXTRA

A number of school systems are taking advantage of our NLX discounts for multiple subscriptions mailed to a single address. Here is a listing of the discounts available for addresses in the United States. Persons ordering from other countries should write for additional information.

Number of subscriptions to a single address	Discount
10 - 19	10%
20 - 39	20%
over 40	30%

The following institutions offer introductory Logo courses for the summer session.

Fairfield University, Attn: Dr. John Schurdak, North Benson Road, Fairfield, CT 06430. (203) 255-5411.

Fresno Pacific College, Attn: Dr. Arthur J. Wiebe, 1717 South Chestnut Avenue, Fresno, CA 93702.

Millersville University, Attn: Dr. Edward Beardslee, Math and Computer Science Department, Millersville, PA 17551. (717) 872-3781.

National College of Education, Attn: Dr. Sandra Turner, 2840 Sheridan Road, Evanston, IL 60201. (312) 256-5150 x 418.

Northern Arizona University, Attn: John W. Glenn, Box 4117, Flagstaff, AZ 86011. (602) 523-4212.

University of Alaska, Attn: Charles K. Ray, 311 Eielson Building, Fairbanks, AK 99701. (907) 474-7021.

University of Houston, Attn: Dr. Sue Brown, 2700 Bay Area Boulevard, Houston, TX 77058. (713) 488-9494 and University of Houston, Attn: Dr. Barbara Sadowski, College of Education, Houston, TX 77004. (713) 749-7103.

Webster University, Attn: Dr. Andrea Rothbart, 470 East Lockwood, Webster Grove, MO 63119. (314) 968-7490.

Instructional Strategies in Logo, a 3 quarter hour course, is offered by Oregon State University during the summer session. For more information, write Dr. Margaret Moore, Weniger 253, Oregon State University, Corvallis, OR 97331. (503) 754-4031.

Come to the University of Virginia this summer for Logo courses. Glen Bull and Paula Cochran teach a 1 credit hour introductory Logo course June 12 - 18, and a 3 credit hour course on using Logo in the content areas, June 19 - July 11. For more information, write to Glen Bull, University of Virginia, 109 New Cabell Hall, Charlottesville, VA 22901. (804) 924-7107.

Dan and Molly Watt and Tony Stavely present their 1985 Logo Institute at Keene State College in New Hampshire July 1 through 19. This popular program of study is designed for educators at all levels of Logo experience. Both theory and applications are studied through a variety of activities. Graduate credit is available. This is one of the more exciting summer Logo opportunities available. For a free brochure, write to Howard Croteau, Summer School Director, Keene State College, Keene, NH 03431. (603) 352-1909.

Logo Computer Systems Incorporated (LCSI) announces two series of Logo workshops in the New York City area. The first concentrates on Logo and the curriculum, and is presented on Thursday afternoons at Packer Collegiate Institute in Brooklyn, starting April 11 and running through May 30. The second deals with Sprite Logo, and meets Fridays at the Convent of the Sacred Heart in Manhattan's East Side from April 12 through May 31. For more information, contact Michael Tempel at LCSI, 555 West 57th Street, Suite 1236, New York, NY 10019. (212) 765-4780.

Two graduate courses on Logo and early childhood will be taught at Oakland University (MI) this summer. An introductory course and one for experienced Logo users both begin in late July. For more information, contact Logo Information For Teachers (LIFT), PO Box 5396, Plymouth, MI 48170.

A summer course entitled Logo Tools for Learning Mathematics is offered by Carleton College. For more information, write Office of Summer Academic Programs, Carleton College, Northfield, MN 55057. (507) 663-4301.

Logo Information For Teachers (LIFT) offers active membership to anyone interested in support for Logo teachers and teacher trainers. Members receive a membership directory, a listing of Logo research projects by members, and quarterly updates. To join, send $10.00 to LIFT, PO Box 5396, Plymouth, MI 48170.

The Educational Computer Consortium of Ohio (ECCO) announces its spring Logo conference to be held Friday, May 3, at Mayfield School in Cleveland. Featured on the program is Joyce Tobias, presently associated with Seymour Papert's Microworlds project. An extensive schedule is planned, with activities for Logo users from beginner to expert. For more information, write to ECCO Logo Conference, 1123 S. O. M. Center Road, Cleveland, OH 44124, or call (216) 291-5525.

The 3rd annual Logo Information For Teachers (LIFT) conference is scheduled for Friday, May 3, at the University of Michigan, from 8:30 a.m. to 4:00 p.m. Co-sponsored by the U of M's Professional Development Office, the conference features Tony Stavely and other nationally known Logo workers. For more information, write LIFT Conference, PO Box 5396, Plymouth, MI 48170.

"Logo and Problem Solving," a 3 credit graduate course, is scheduled to be taught at Wright State University in Dayton, OH, beginning June 10. The course provides an introduction to Logo with a major focus on the problem solving processes of the language. Dr. Carl Benner is the instructor. For more information, write Bonnie K. Mathies, Program Coordinator, 244 Millett Hall, Wright State University, Dayton, OH 45435, or call (513) 873-2509.

Logo and the Learning Environment, a 4 quarter hour course, is offered this summer at Eastern Washington University. For more details, write Stanley Robinson, Department of Computer Science, Cheney, WA 99004. (509) 458-6260.

The Graduate School of Education of Lesley College has scheduled several Logo courses for the summer session. Harriet Deane, Lesley College, 29 Everett Street, Cambridge, MA 02238. (617) 868-9600 x 360.

A course on advanced Logo will be taught at Long Island University this summer. For details, write Patricia Devlin, Department of Educational Technology, Long Island University, Greenvale, NY 11548. (516) 299-2199.

Looking for a course in advanced Logo for this summer? Contact Dr. Richard Forcier, Educational Technology Program, Western Oregon State College, Monmouth, OR 97361. (503) 838-1220 x 441.

Purdue University has scheduled a course entitled Logo and Elementary Computer Graphics for this summer. Write Dr. James Lehman, Department of Education, Purdue University, West Lafayette, IN 47907. (317) 494-5670.

Teachers College of New York City offers a course called Teaching Logo this summer. For details, contact Howard Budlin, Teachers College, Box 8, 525 West 120th Street, New York, NY 10027.

Courses in Logo for the beginner, the intermediate, and the expert are scheduled this summer at the University of California at Riverside. For more information, write to Sue Teele, University Cottage, University of California at Riverside, Riverside, CA 92521. (714) 787-4361.

The University of Montana offers two Logo courses this summer: Turtle Graphics for Teachers, and Logo in Geometry for Teachers. Contact Johnny Lott, Department of Mathematical Sciences, University of Montana, Missoula, MT 59812. (406) 243-2403.

Logo in Education, a 4 quarter hour course, is offered by the University of Oregon, June 25 - August 16. Contact Fred Beisse, Department of Computer and Information Science, University of Oregon, Eugene, OR 97403. (503) 686-4408.

Gold Disk is featuring special computer offers good until May 31, 1985. They include Wabash SS SD disks, box of 11, $12.90, Wabash SS DD disks, box of 10, $14.90, Terrapin Logo, $74.95, Fontrix $49. Gold Disk, 24 East Pennsylvania Avenue, Bel Air, MD, 21014. (800) 368-2260.

Single sided double density disks for $1.00 each in any quantity are offered by Blackship Computer Supply. This is about the lowest price we have seen. Also listed are double sided double density disks for $1.39 each. Please add $3.00 for shipping and handling. Blackship Computer Supply, PO Box 883362, San Francisco, CA 94188. (415) 550-0512.

BEST COPY AVAILABLE

From the Editor

Logo and Sex

Seymour Papert observed that nearly half of the attendees at the Logo 84 conference last year were women. To an outsider, this might come as a surprise, since the computer field seems to be dominated by males. However, I feel that the large number of women was a manifestation of the Logo philosophy in action. "No threshold, no ceiling" should apply to everyone.

But is it really that way in the Logo classrooms?

The sexism issue is an important one. I recently took a workshop which sensitized me to how teachers react in different ways to male and female students.

One of the things suggested by the workshop instructor was that teachers tend to give detailed instructions to male students, but operate equipment or perform tasks for the female students, thus denying them the opportunity to learn by doing.

Since the workshop, I have been more aware of the impact of my words and actions in class. For example, I noticed that I would sometimes type at the Logo keyboards of the female students instead of giving them instructions. That was changed in a hurry!

I have asked my students to help me identify any other similar actions which put either males or females at a disadvantage.

Perhaps it would be worthwhile for all of us to "observe ourselves" as we interact with our male and female Logo students. Do we treat them differently? Do we allow access or deny opportunity? Do we encourage or discourage?

Here are a couple of recent articles well worth reading.

"The Computer Gender Gap," by Gordon Williams, Family Weekly magazine (Sunday newspaper insert), February 24, 1985, p. 7.

"Sexism in the Schoolroom of the '80's," by Myra and David Sadker, Psychology Today, March 1985, p. 54.

For a more comprehensive look at the equity issue as a whole, see The Computing Teacher of April 1984.

Let's make it "no threshold, no ceiling for everyone."

FD 100! Tom

Tipps for Teachers

by Steve Tipps

Survey Sense II

When information is collected with a survey, questionnaire, or test, organizing and analyzing are important tasks. Collecting data, displaying it on a chart, finding the items with the most or fewest responses, and calculating the percentages of different categories have been discussed for the last two months. This month, data organizing tools are expanded to the high and low scores and the average. The idea of the "middle" of the information and its "spread" are also introduced.

MAKING ALLOWANCES

Interesting surveys need interesting topics. Money is always interesting to most people. How do you get it? How much do you get? What do you do with it? Doing a survey on allowances follows the same process as a face-to-face interview.

How much allowance do you get each week?
(Wait for an answer.)
Thank you for answering.

The program for this dialog is fairly simple, although a few extra directions help you get the right information. The tricky part is forming the list of allowances. This survey puts all the answers into the "ALLOWANCES list.

```
     TO SURVEY
     PRINT  [ HOW MUCH ALLOWANCE DO YOU  GET
EACH WEEK?]
     PRINT [TYPE YOUR ANSWER IN DOLLARS  AND
CENTS]
     PRINT [WITHOUT A DOLLAR SIGN.]
     PRINT [ THIS IS AN EXAMPLE:]
     PRINT [ 2.50 ]
     PRINT [ NOW TYPE IN YOUR ALLOWANCE.]
     MAKE  "ALLOWANCES  LPUT FIRST  READLIST
:ALLOWANCES
     PRINT [ THANK YOU FOR ANSWERING.]
     END
```

At the beginning, the global variable "ALLOWANCES must be an empty list. This is only done once before any surveys are made.

```
     MAKE "ALLOWANCES [ ]
```

Tipps continued

Each SURVEY will add one item to the list. Of course, you could make separate surveys for different classes by defining separate procedures and variables names for each class.

```
MAKE "BLITMAN.ALLOWANCES [ ]
TO SURVEY.BLITMAN.CLASS
etc.
```

Survey forms can be as fancy as you want. On the allowance survey, a title page with money or a piggy bank is a nice addition. A more complete allowance survey might even ask for a distribution of the total spent into several categories: snacks, records, videos, movies, clothes, etc. Each of the categories could be stored in separate lists.

Now you want to organize, summarize, and display to show how much allowance is typical of the people in the survey. A couple of calculating tools can very handy.

LOOKING FOR THE EXTREMES

How can the list of allowance values be described? One way is to use the highest and lowest allowance. Here is a sample survey list. Look through the list and pick out the high and the low amounts. (Although amounts such as 3.00 were typed in when the SURVEY was taken, most Logo versions eliminate the zeroes when defining the ALLOWANCES list.)

```
     ALLOWANCES IS [1.  2.5 1.5 2.25  4.  1.
1.25 3. 2.5 1.25 2.5 3. 2.75 2. 2.75 3. 2.25
2.5 3. 2.5 2.25 3. 2.75 2.5 3.]
```

Looking for the highest, you would probably say to yourself, "The first one is a dollar; that's the highest so far. The next is 2.50; that's higher. Now I'll use 2.50 to compare with the next one." The highest is selected by picking and comparing. Each time a higher value is found, it replaces the former high value.

Talking through this scanning process helps to understand how Logo could go about finding the high and low, because the process parallels the talk.

```
TO FIND.HIGH :LIST
IF :LIST = [] [ OP :HIGH ]
TEST :HIGH < FIRST :LIST
IFTRUE [ MAKE "HIGH FIRST :LIST ]
OUTPUT FIND.HIGH BUTFIRST :LIST
END
```

To begin, establish a low number as the first "HIGH value.

```
MAKE "HIGH 0
PRINT FIND.HIGH :ALLOWANCES
4.
```

The procedure to FIND.LOW is similar, only the initial value for "LOW must be set high to start.

COUNT FOR COUNT

Another way to organize the information is to count the number of times each value appears in the list. If you were doing this by hand, you would start at the low value and count how many there were. Then you would move to the next higher value and count again. You would keep doing this until you got to the highest value. That is exactly what Logo can do, too.

```
     TO COUNT.EACH :LIST :TARGET
     IF :TARGET > :HIGH [STOP]
     MAKE WORD "C :TARGET 0
     TYPE :TARGET TYPE [\ \ \ ]
     PRINT COUNT.EACH.LOOP :LIST
     COUNT.EACH :LIST :TARGET + 0.25
     END

     TO COUNT.EACH.LOOP :LIST
     IF  :LIST = [] [OUTPUT THING  WORD  "C
:TARGET]
     TEST :TARGET = FIRST :LIST
     IFTRUE  [MAKE  WORD "C  :TARGET  (THING
WORD "C :TARGET) + 1]
     OUTPUT COUNT.EACH.LOOP BUTFIRST :LIST
     END
```

The new "trick" in these procedures is to create a variable name for counting each value. WORD "C :TARGET makes a variable name such as "C1.5 for counting the 1.50 values. The results look something like this. Use a cursor command such as SETCURSOR LIST 8 LAST CURSOR just before the PRINT line in the COUNT.EACH procedure if you want to have the counting results appear in a neat vertical line.

```
COUNT.EACH :ALLOWANCES :LOW
1.   2
1.25   2
1.5   1
...
...
4.   1
```

Tipps continued

A PICTURE IS WORTH A THOUSAND CHARTS

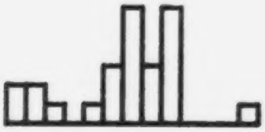
A graphic display would show even better the distribution of the allowance values. A BAR chart is easy to plot using values from COUNT.EACH.LOOP in the SHOW.EACH procedure.

```
TO SHOW.EACH :LIST :TARGET
IF :TARGET > :HIGH [STOP]
MAKE WORD "C :TARGET 0
BAR 5 *  COUNT.EACH.LOOP :LIST
MOVE
SHOW.EACH :LIST :TARGET + 0.25
END

TO BAR :SIZE
REPEAT 2 [ FD :SIZE RT 90 FD 5 RT 90]
END

TO MOVE
RT 90 FD 5 LT 90
END
```

The bar chart looks like this.



The picture shows how how far the allowances are spread out and indicates the the most frequent allowances. It also gives a visual sense of the middle value. Some Logo versions make it possible to label the values on the graphics screen.

MIDDLING

When information is collected in categories equal in size, one way to summarize the data is with the middle scores. Actually, there are two different middle scores which can be used in certain circumstances. You may remember them from your Tests and Measurement courses.

The median is the score identified by counting halfway from the bottom to the top of all the scores. A median can be found quickly. For example, if the scores are [12 14 14 14 15 17 18 20 20], the median is 15 because it is the 5th of 9 scores.

The more common way to describe the middle is with the mean. The mean is the arithmetic average of scores calculated by adding all the scores together and dividing by the number of scores. The total of the nine scores above is 146, so the mean is 16.

Older children are usually familiar with averages from averaging grades. Although a calculator is useful for determining averages, Logo can be used for both collecting and calculating information. Its advantage is that information can be stored easily in the computer in lists and be subjected to several different analyses. Many calculators have no memory for data storage, or ways of keeping several different sets of information.

Writing a program to calculate the mean is fairly straightforward. Work through the list adding one item at a time to the total. Then divide the total by the number of items in the list.

```
TO AVERAGE :LIST
MAKE "SUM 0
OUTPUT ( TOTAL :LIST ) / COUNT :LIST
END

TO TOTAL :LIST
IF :LIST = [] [ OUTPUT :SUM]
MAKE "SUM :SUM + FIRST :LIST
OUTPUT TOTAL BUTFIRST :LIST
END

PRINT AVERAGE :ALLOWANCES
2.4
```

The way you handle the average procedure in your class will depend on the purposes of your lesson. For example, if you are working in sixth grade through high school mathematics, you can have students describe how to calculate the average and help them express the process in Logo. If you are using the procedure as a tool for a data collection, you can write the tool in a file for children to use. Whether to teach how to write the program or use it for analysis is not crucial; your goal for students is the important focus for your Logo work.

The bar chart provides a picture of how spread out the scores are. With a different survey you might get a different distribution. Looking at the spread is a very important idea with data. The standard deviation is a way of describing the spread of the scores. Next month, the calculation of the standard deviation will be introduced. For now, I'll leave it as a challenge for you and I'll share my method with you in May.

Tipps continued

(By the way, the conversion of these procedures to Terrapin is straightforward. However, there is some kind of a rounding error present which fails to select all categories in COUNT.EACH. One way around this is to collect the allowance information in pennies to the nearest 25c. For example, 2.50 should be 250.)

Steve Tipps is the West Professor of Education at Midwestern State University in Wichita Falls, TX. He presents Logo workshops for school systems throughout the United States.

PAWS for the Turtle

Harold Nelson contributes this program for those whose Logo versions do not have a PAUSE command. Embed PAWS in your favorite procedure and study how it works. What limitations can you find?

```
TO PAWS
TYPE [PAUSING... ?]
PAWS.AUX READLIST
END

TO PAWS.AUX :INSTRUCTION.LIST
IF :INSTRUCTION.LIST = [CO] [STOP]
RUN :INSTRUCTION.LIST
TYPE [PAUSING... ?]
PAWS.AUX READLIST
END
```

For more suggestions on using PAWS or PAUSE for debugging, see Harold's article on page 1 of this issue.

Looking for Freebies?

Freebies is a magazine no teacher should be without! Published every month with a circulation of over 500,000, Freebies contains information on how to obtain free (or nearly free) items. Each issue has a special section for teachers. The cost? A mere $6.97 per year. By the way, the NLX UVA Logo Class Penpal Network was featured in their March 1985 issue. Freebies, PO Box 20283, Santa Barbara, CA 93120. (805) 962-9135.

NLXionary

A Lectionary and Discussion of Logo Readings

by Griff Wigley

I remind readers that I am opening up this column to you for the purpose of idealogical exchange on issues raised by current articles on Logo. I'll reserve my comments on the selections until I hear from you. Write to me at 918 College Street, Northfield, MN 55057.

"Logo and the Teaching of Problem Solving: A Call for a Moratorium," by Toby Tetenbaum and Thomas Mulkeen, Educational Technology, November 1984.

>The authors note that the excitement generated by Logo counts not so much from from its perceived value as a programming language, but rather from its promises to enhance children's cognitive capacities and develop higher level congitive skills which will generalize or transfer to other content areas. They challenge this view, comparing those who believe Logo teaches children how to think to those who claim that the study of Latin improves the mind through the discipline required to learn it. Research has failed to prove that there is any significant transfer from training on one task to success on another.

>They challenge the idea that the type of thinking employed when one solves a problem can be improved if one learns to "think" like a computer does when it solves a problem. They contend that the whole arena of problem solving is so huge and complicated that it is impossible to determine a single set of skills that can be identified as crucial to problem solving ability. They cite research that indicates problem solving can't be separated from subject matter content. They contend that Logo proponents have created unrealistic expectations regarding its ability to foster a generalized problem solving ability.

>Tetenbaum and Mulkeen refer heavily to the Bank Street study by Roy Pea and associates which indicated children who learned Logo didn't preplan prior to programming, didn't understand features of the language

NLXionary continued

(recursion, conditional statements, sequentiality) even though they used them, and failed to display better planning skills than the control group.

>Rather than throwing out Logo, the authors suggest a moratorium on promoting it as a generalized problem solving model. They indicate that it might yet prove to be so, if educators create two conditions that enhance the transfer of knowledge and skills:

1. Some of the processes or knowledge used in one task should be identical to those learned while engaging in another.
2. Learners should be made aware specifically of the skills involved in the task, as well as the specific content of the task.

>Pea has evidently suggested that instruction in thinking skills in conjunction with Logo might be beneficial, despite his description of Logo as "Wheaties for the mind." The authors, however, clearly doubt if teaching problem solving through the teaching of programming is the most efficient way to bring it about.

>"Before tens of thousands more children are taught Logo, it seems advisable to give serious consideration to its purpose."

>Do we all agree?

(Ed. Note: If you have access to CompuServe, you may find the lively discussion about this article on TeleCUE to be of interest. Type GO WCC-21 after logging on, and read Logo messages from #1953 to #2052.)

Griff Wigley is a co-owner of Family Computing, Inc., a facilitator in the Faribault (MN) Public Schools, and a school board member of Prairie Creek Community School.

TurtleTips

by

Donna Lanyi
and Jane Toth

By this time of the year, your students are creating some interesting procedures. Many of their ideas are being saved on disks. Sharing is a great way to motivate students. Let's take a look at ways to find time for students to share their ideas with their classmates.

TEACHER HINTS

Planning a 20 - 30 minute sharing time per week is an excellent habit. This sharing time can be used for multiple purposes.

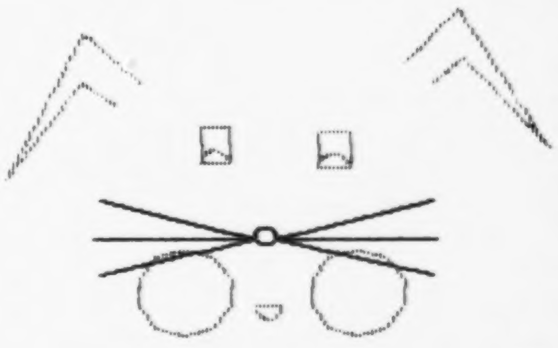
First, it enables the class members to see what others are doing. When a student completes a Logo project, your classroom doesn't have to be interrupted for "show and tell." Everyone knows they will get to see the project during sharing time.

Second, the sharing period can be used as a catalyst for other students who have had "idea stagnation." We have seen a number of students get exciting ideas for spinoff projects from something a classmate reports.

A third purpose of a sharing time is to allow students an opportunity for constructive criticism of each other's work. The teacher should direct the discussion to keep it positive. As a result of these remarks, additional improvements, additions, and extensions often suggest themselves.

PROCEDURE OF THE MONTH

Hop to it! Have your class design an Easter parade of spring procedures. Use our bunny procedure as a motivating idea.



If you would like a free listing of this procedure, send us a self addressed stamped envelope.

We invite elementary teachers and students to send procedures or ideas to be shared. Donna Lanyi and Jane Toth, Beall Avenue School, 716 Beall Avenue, Wooster, OH 44691.

Donna Lanyi and Jane Toth use Logo in their elementary classrooms in the Wooster (OH) City Schools.

Listful Thinking

by
Glen Bull and Paula Cochran

Word Machines

Spring has finally arrived at Mr. Jefferson's University. Mr. Jefferson was something of a tinkerer, continually designing gadgets and architectural improvements for Monticello. We have been tinkering with Logo machines.

LOGO MACHINES FOR LANGUAGE TINKERING

Judah Schwartz, co-director of the Educational Technology Center at Harvard, compares Logo to a "machine shop" in which any tool can be built, but only with considerable time and skill (see NLXionary, March NLX, 1985). For the past year, we have emphasized use of Logo tools to teach content-specific concepts in language arts.

In this month's column, straight from our Logo machine shop, we present the Great American Word Machines. Schwartz says (and we agree) that using Logo can be like trying to make a pair of pliers from a blank piece of metal. In this column we present some intermediate tools. These tools let you stock your own machine shop with pliers, screwdrivers, and other tools. We hope this lets you begin teaching language arts, instead of having to spend time making Logo tools to teach language arts.

Children are inquisitive about language. A four-year-old may experiment with variations of a word in a sing-song voice. Children sometimes invent their own "secret" languages. Piglatin is an example which children re-invent in each generation. This early fascination with language is rarely incorporated into formal grammar instruction in the classroom. We need ways to use children's natural playfulness and enthusiasm for language in English class.

By way of contrast, here is a typical lesson from a sixth grade English text:

> In the sentences below, change each adjective to an adverb. Tell how each adverb is formed.

1. Tom was _______ answering his NLX mail.
(eager)

2. Steve was __________ writing his column.
(quick)

The object of the lesson is to learn how adverbs are constructed. By adding -ly endings, the student is supposed to deduce a rule about English adverbs. How can we construct a Logo machine which will supplement and extend this lesson?

WHAT'S IN THE MACHINE SHOP?

Our Logo machine shop comes equipped with four types of tools.

1) separators
2) joiners
3) omitters
4) testers

These tools are used to build Logo machines, for the exploration of different subjects. We're going to look at some word machines.

The Logo command WORD joins any two parts:

```
PRINT WORD "EAGER "LY
EAGERLY
```

Here's a Logo machine which uses the command WORD to convert adjectives into adverbs.

```
TO ADVERB :ADJECTIVE
OUTPUT WORD :ADJECTIVE "LY
END

PRINT ADVERB "EAGER
EAGERLY
PRINT ADVERB "QUICK
QUICKLY
```

CAUTION!
WORDS UNDER CONSTRUCTION

There are several ways to use this Logo machine to supplement the textbook assignment. The teacher could ask students to guess what the machine is doing. In this case, the children are being asked to deduce the rule which the machine uses to create adverbs. This could be a group activity in which the teacher types in sample words. Older students could be given a page of sample words to test on their own. A page of sample words from the assignment might look like this:

IN	OUT
eager	eagerly
soft	________
quiet	________
sure	________
sudden	________

Listful Thinking continued

Another way to use this machine would be to give children the rule, then let them explore the boundaries of the rule using the Logo machine. The assignment might look like this:

> Here is a rule for making adverbs from adjectives:
>
> Add -ly to some adjectives to form an adverb.
>
> slow --> slow + ly = slowly
>
> Use the Logo ADVERB machine to test this rule. Which of these words make correctly formed adverbs, and which do not?

IN	OUT
nice	nicely
abrupt	________
frantic	________
sincere	________
think	________

TINKERING AND TESTING

It's possible to turn the idea around, and construct a Logo machine that checks to see if the last two letters are -ly. This machine uses a Logo tool that separates the last two letters from the word. (See the end of the column for LAST.2.LETTERS.)

```
PRINT LAST.TWO.LETTERS "NICELY
LY
PRINT LAST.TWO.LETTERS "FROG
OG
```

The LAST.2.LETTERS procedure is used here to construct a machine that tests for adverbs ending in -ly.

```
TO ADVERB? :ITEM
OUTPUT (LAST.2.LETTERS :ITEM) = "LY
END
```

It works like this:

```
PRINT ADVERB? "NICELY
TRUE
PRINT ADVERB? "FROG
FALSE
```

Of course, the machine won't work perfectly.

```
PRINT ADVERB? "SALLY
TRUE
```

A child will often discover these cases, and ask why the machine doesn't work. This provides a good opportunity for a discussion about the kinds of knowledge people need to make distinctions between nouns and adverbs. What else would the ADVERB? machine need to work better?

THE ALL-PURPOSE CHECKER CHECKS ADJECTIVES

The same approach used for the "ly" checker can be used to build a general purpose tester. This checker could be used to test adjective suffixes, for example.

First, make a list of adjective suffixes. We use "ful" and "ive" but you could make up your own list.

This procedure outputs the list of adjective suffixes.

```
TO ADJ.SUFFIX
OUTPUT [FUL IVE]
END
```

The procedure to test for adjective suffixes looks like this. The LAST.3.LETTERS procedure is at the end of the column. (If you are using Terrapin Logo, use the CHECK procedure at the end of the column.)

```
TO CHECK :ITEM
TEST MEMBERP LAST.3.LETTERS :ITEM ADJ.SUFFIX
IFTRUE [OUTPUT "TRUE]
OUTPUT "FALSE
END
```

This particular checker is used like this:

```
PRINT CHECK "THANKFUL
TRUE
PRINT CHECK "NICELY
FALSE
PRINT CHECK "MASSIVE
TRUE
```

This suffix checker looks for adjective suffixes, so it picks out "thankful" and "massive", but not the adverb "nicely." The suffix tester could be used in an assignment like this:

> Here is a list of words. Test each word with the Logo machine CHECK. Then write down the result in the blank beside the word.

Listful Thinking continued

IN	OUT
thankful	True
kindness	________
massive	________
nicely	________
darkness	________
permissive	________
bountiful	________

After checking each word, decide what kind of word the suffix tester is identifying: NOUN, ADJECTIVE or ADVERB.

These ideas are designed to enhance traditional pencil-and-paper language activities. Logo word machines can help students explore language and examine rules by taking advantage of their natural curiosity. Next month we will look at some other Logo language machines that use separators, joiners, omitters, and testers.

PROCEDURES

```
  TO PIGLATIN :ITEM
  OP ( WORD BUTFIRST :ITEM FIRST :ITEM "A )
  END

  TO LAST.2.LETTERS :ITEM
  OP WORD NEXT.TO.LAST :ITEM LAST :ITEM
  END

  TO NEXT.TO.LAST :LETTER
  OP LAST BUTLAST :LETTER
  END

  TO LAST.3.LETTERS :ITEM
  OP WORD THIRD.FROM.LAST :ITEM
LAST.TWO.LETTERS :ITEM
  END

  TO THIRD.FROM.LAST :LETTER
  OP LAST BUTLAST BUTLAST :LETTER
  END
```

If you are using Terrapin Logo for the Apple, or Commodore Logo, use this CHECK procedure.

```
TO CHECK :ITEM
TEST MEMBER? LAST.3.LETTERS :ITEM ADJ.SUFFIX
IFTRUE OUTPUT "TRUE
OUTPUT "FALSE
END
```

If you are using Version 1 of Terrapin Logo for the Apple, you will also need the MEMBER? procedure below.

```
TO MEMBER? :ITEM :LIST
IF :LIST = [] OUTPUT "FALSE
IF :LIST = "  OUTPUT "FALSE
TEST :ITEM = FIRST :LIST
IFTRUE OUTPUT "TRUE
OUTPUT MEMBER? :ITEM BUTFIRST :LIST
END
```

Glen Bull is a professor in the University of Virginia's School of Education, and teaches Logo courses at both the graduate and undergraduate level. Paula Cochran is a reformed English major who studied linguistics at the University of Cambridge. She is now a speech-language pathologist working with language-disabled children and Logo.

Logo Notes

The following Logo versions are available from The Programmer's Shop. Dr. Logo at $119, Experlogo for $140, Commodore Logo for $65, TI Logo at $99, PC Logo for $135, Terrapin Logo for $89, and TLC Logo for $95. The Programmer's Shop, 128 Rockland Street, Hanover, MA 02339. (800) 421-8006.

Available from Triton Products: Early Logo Learning Fun (#AAFH), a TI 99/4A cartridge for preschool children by Spinnaker at $11.95, and the TI Logo II (#AEBE) cartridge for the TI 99/4A at $99.95 plus $3.00 for shipping. A catalog listing many other TI 99/4A products is also available. Triton Products, PO Box 8123, San Francisco, CA 94128.

Miles Kimball Company, a mail order firm, sells a "cast iron" turtle string holder complete with a large ball of red twine. We threaded the twine so it came out of the bottom of the turtle, and used it as a concrete drawing object. Thumbtacks fix the string in place as the turtle maneuvers over a cardboard surface. Item # 7906, String Holder Turtle, $3.49 plus $1.95 (ouch!) shipping. Miles Kimball Company, 41 West 8th Avenue, Oshkosh, WI 54906. (414) 231-3800.

Logo Disserts:

Dissertations Dealing with Logo

by Barbara Elias

What aspects of Logo instruction may serve to prepare children for our more technically oriented society? Problem solving, motivation to learn, and attitudes were investigated in a study of the implementation of a Logo programming course for fourth grade children. Eight of fifteen local fourth grade classes were used as subjects, while the remaining seven served as the controls.

Classroom teachers who provided the instruction in Logo were trained in a summer inservice Logo programming course at the University of Virginia. The course placed special emphasis on the Logo philosophy, and stressed the importance of encouraging exploration with Logo. It incorporated such concepts as modularity and debugging.

Once school began and classes started, teachers were expected to model course behaviors for the children involved. Teacher inservice continued throughout the school year and the inservice instructors worked closely with the teachers in developing classroom activities and curriculum ideas.

Teacher implementation of the Logo course was expected to vary because teachers were free to choose the most appropriate means for their individual classrooms. Generally, children worked in pairs during the 45 hours of class time set aside for the computer in the year long study.

The following three tests were administered at the beginning and at the end of the school year: the Cognitive Abilities Test (Nonverbal form), the Fennema-Sherman Mathematics Attitudes Scales, and the Scale of Intrinsic versus Extrinsic Orientation in the Classroom.

Results indicate that the Logo programming experience had a positive influence on problem solving abilities and on attitudes toward learning mathematics. Even though the differences were small, they represent potentially important differences between children who have had Logo programming and those who have not. While these results look promising, it is recommended that this study be replicated and extended, that Logo be integrated into the mathematics curriculum, and that the importance of continuing teacher inservice support to the success of a Logo project be recognized.

>>>Evans, Jr., H. R. (1984). Some Effects of Logo Programming Instruction with Fourth Grade Children. (Doctoral dissertation, University of Virginia, 1984).

(Ed. Note: Each month, Barbara Elias highlights a dissertation or thesis dealing with Logo. She is accepting copies of recent research results for the NLX Dissertation and Thesis Repository. Reports may be mailed to her in care of NLX, PO Box 5341, Charlottesville, VA 22905.)

Barbara Elias is an assistant professor in the Education Department of Virginia State University in Petersburg, VA, and a doctoral candidate at the University of Virginia.

Logo Announcements

Terrapin is introducing an upgrade for their Apple computer version of Logo. Terrapin Logo 3.0 is really two languages in one. When booted on an Apple computer, the program identifies the model and amount of resident memory, and then loads either a 64K version, or one which can address a full 128K, if available. Thus, with a single Logo disk, you can take advantage of the full memory capabilities of every computer in the Apple II family. In addition to addressing more memory, Terrapin 3.0 also loads much faster than previous versions. The suggested list price is $99.95. Owners of Terrapin Logo 1.3 or 2.0 may upgrade for a small fee. For more information, write to Bob Paolucci, Terrapin, 222 Third Street, Cambridge, MA 02142, or call (617) 492-8816.

Terrapin also announces the release of a Logo cartridge for the Commodore Plus/4, making available roughly twice the memory space of Commodore Logo at a suggested list price of $74.95.

ExperTelligence Inc. announces ExperLogo, Logo for the Macintosh. Included are many advanced features not found in other Logo versions. Additionally, ExperLogo is compiled, resulting in fast programs. The suggested list price is $149.95. ExperTelligence, 559 San Ysidro Road, Santa Barbara, CA 93108. (805) 969-7874.

Teacher to Teacher

The Turtle Goes to Kindergarten

A guest review

by Jan Cardwell

Turtles are popping up everywhere! Since Logo went public, turtles have gained rapidly in popularity and in possibility. The momentum of the race continues forward as turtles are propelled into the kindergarten and introduced to the very young.

What better environment could there be for the Logo turtle than the kindergarten? Renown for their curiosity and willingness to explore, kindergartners mirror Seymour Papert's vision of Logo in Mindstorms.

For educators who have not had the "free" time to familiarize themselves with Papert's work, Donna Bearden and Kathleen Martin have the answers in a clear, concise, and easy to read book, The Turtle Goes to Kindergarten. Instead of bringing the student up to the level of the computer, Donna and Kathleen have prepared ways to bring Logo to the level of the kindergartner.

LEVELS OF LOGO

In order to broaden the Logo capabilities of such beginning readers, the authors have developed programs for single stroke commands. The three levels presented increase in degree of difficulty, and correlate to a kindergartner's interest and developmental abilities.

Realizing that young children learn best by doing, the authors have included activities which help the child experience abstract concepts in a very concrete way.

Especially inspiring is the section entitled "Pre-Logo Activities." Here, suggestions are made which immerse the kindergartner in learning. Children become turtles, making discoveries as turtles. These Logo lessons are not forgotten when the students are guided by self discovery.

WHERE'S THE BEEF?

My favorite activity was one of the "Playing Turtle" games. Students take turns pretending to be the turtle, as classmates give verbal commands directing the turtle to — the turtle food!

All kindergarten teachers know what a motivator this can be! What would kindergarten turtles eat? When I asked my students, they responded with answers ranging from flies (yuk!) to jelly beans!

Donna and Kathleen certainly know the way to a kindergartner's heart (and stomach) as well as the way to the brain!

Throughout the book, teachers are encouraged to integrate Logo learning into the kindergarten classroom environment. In fact, the entire second part of the book is devoted to programs and techniques which teach basic kindergarten curriculum concepts (spatial relations, colors, arithmetic) through the use of Logo and the computer. The programming work has already been done.

IN SUMMARY

In conclusion, I must mention that the most compelling aspect of The Turtle Goes to Kindergarten is the way the authors encourage the reader to expand upon the ideas presented. Not only do Donna and Kathleen teach us how to incorporate Logo into the kindergarten, but they also inspire us to do some learning, too!

The Turtle Goes to Kindergarten is available from the authors for $5.00 plus $1.00 shipping and handling. Martin - Bearden, Inc., PO Box 337, Grapevine, TX 76051.

Jan Cardwell, a former preschool teacher, is a program coordinator for the Science Museum of Virginia. She has published several Logo teaching tips in national educational periodicals.

Free Logo Materials

The Cotting School for Handicapped Children in Boston has a limited number of free booklets on Logo activities available. Logo for Young Children and Children with Cognitive or Physical Handicaps: Simplified Versions of Logo by Susan Jo Russell offers four modified Logo versions and suggestions for their use. Logo Games and Activities by Susan Jo Russell and Eileen Kiernan includes games and other activities for 6 to 10 year old children, and for children with special cognitive or physical needs. Send a request on your letterhead to Development Office, Cotting School for Handicapped Children, 241 St. Botolph Street, Boston, MA 02115. This offer is good only while the supply lasts.

前 100

by
Hillel
Weintraub

Last month, I introduced the Bynas Division of UNY Company and their work in the spread of Logo throughout Japan, and told of their Logo Journal, a monthly newsletter devoted to keeping a growing Japanese Logo community informed about the educational and commercial activities of Logo around the world. This month, I would like to introduce a second project presently under development by Hiroyoshi Goto of Bynas: the formation of the Japanese Logo Research Group, an organization composed mainly of educators which has been involved in Logo developments in Japan.

The first work of the group was to develop a set of standard Logo specifications for use with Japanese primary school students. The tentative list of specifications will be announced this May, and the final set in December. The purpose of creating such a list is to "make Logo easily accessible so that programs can be accumulated and shared." In order to give Logo the ability to operate on several types of machines, this group is planning to make their language specifications public and encourage all Logo producers to adopt it.

In my experience using Logo (which is almost totally with 10 - 12th graders working with words and lists), I have developed some strong feelings about useful primitives, hardware capabilities, and the like. Here are a couple of the ideas I plan to share with the research group.

1. A TRACE command is extremely important. A primitive is easier to use than a TRACE procedure which needs to be defined by the user. One suggested improvement would be to have the program text output printed on the screen in inverse letters. This would make it easier to separate what is part of TRACE and what is output of the procedure in operation. Even better would be to have the TRACE output shown on a separate "page" especially for debugging. Thus, the TRACE instructions could be overlaid without disturbing the actual program operation. This would emphasize the importance of planning and debugging in Logo. (Ed. Note: For a different perspective on TRACE, see Harold Nelson's article elsewhere in this issue.)

2. Many students do not use the error message fully in the debugging process. I find that my students do not spend much time studying Logo complaints before they type ED. Once they type ED, the message is gone forever. In Japan especially, the compulsion to get rid of errors before they bite you is ingrained in students from the time they can pick up a pencil. It is a fact that some students here write with a pencil in one hand and an eraser in the other! I would suggest an editing system which does either of the following things:

a. Has no separate editor and thus doesn't erase error messages when displaying a procedure for editing.

b. Has an editing mode which displays the error message in a special box somewhere.

c. Has a "bug buffer" which recalls the last error message any time upon pressing a key, even when the user is in the editor.

d. Uses a special debugging "page" (as mentioned in 1 above) which could be overlaid with the error message.

Preservation of the error message would reinforce one of the main teaching points of Logo: to use our mistakes to help us learn!

These are a few of the ideas I plan to present to the Japanese Logo Research Group. How about you? Do you have anything to suggest which I could pass along for them to consider? If so, send your ideas to me at Doshisha International High School, Tatara, Tanabe-cho, Tsuzuki-gun, Kyoto-fu 610-03 Japan.

Hillel Weintraub is a teacher at Doshisha International High School in Kyoyo-fu, Japan, and is the president of the Society for Microcomputing in Life and Education (SMILE).

The New Microquests

Microquests, the popular monthly Logo publication from Kathleen Martin and Donna Bearden, will have a new look in the fall. Kathleen reports that they are changing the format to a quarterly 50-page booklet which will use a project approach to promote thinking skills, facilitate curriculum integration, and provide student encouragement and challenges. Disks will be available for each issue. Donna and Kathleen produce work of the highest quality, and this series of booklets will be no exception. For more information, write Martin-Bearden, Inc., PO Box 337, Grapevine, TX 76051.

Q and A

by
Jim McCauley

Q. I have managed to define an "inaccessible procedure" in Apple Logo:

```
TO "1
FD 100 RT 45 BK 20 FD 20 LT 45 BK 100
RT 90 FD 20 BK 40 FD 20 LT 90
END
```

I can't get it to run! And the only way I can change it in the editor and redefine it with Control-C is to put the quotation mark (") in front of the name (1) each time. What's going on here?

<[-?-]>

A. Usually, when you try to define something in any of the Logo versions from Logo Computer Systems Incorporated (LCSI), one of two things happen. If the name that you choose for the procedure happens to be the name of something that is already defined (either a primitive or a procedure), you get an error message as soon as you type in the "defining line."

```
TO FORWARD :STEPS

FORWARD IS A PRIMITIVE
```

If you choose the name of an already-defined procedure (say its name is FOO), you get:

```
FOO IS ALREADY DEFINED
```

If you choose as a procedure name something that is not yet defined, you are permitted to proceed normally.

OUT OF THE ORDINARY

You have uncovered an unusual situation. Numbers and certain other characters (control characters, for example) have special meanings in Logo, and they are, in a sense, reserved for use by the system.

However, they are not protected against use as procedure names in the way that primitives and defined procedures are.

The Apple Logo Reference Manual mentions "formal Logo" on page V. It is a section that bears scrutiny. In the more normal mode of use, Apple Logo is "relaxed." That is, one can take various shortcuts.

In formal Logo, however, any object that is not quoted or does not begin with a colon (as variable names and values do) is assumed to be the name of a primitive or procedure. Therefore, if we were obliged to program in formal Logo, the favorite saying of this newsletter's editor would appear:

```
FORWARD "100
```

Apple Logo recognizes this command, and carries it out.

If you really want to run your 1 procedure, here is one way.

```
TO RUNIT :WHAT
RUNNER BUTFIRST TEXT :WHAT
END

TO RUNNER :RUN.LIST
IF EMPTYP :RUN.LIST [STOP]
RUN FIRST :RUN.LIST
RUNNER BUTFIRST :RUN.LIST
END

RUNIT 1
```

Whew!

(Ed. Note: If you have a question about Logo programming techniques or the thinking which leads to Logo programming, send it to NLX Q & A, PO Box 5341, Charlottesville, VA 22905. Each month, Jim McCauley answers selected questions in this column.)

Jim McCauley is a Coordinator of Computer Education for the Santa Clara (CA) County Office of Education, and has written Logo articles for many national publications.

Welcome to the French Logo Users Group

The French Logo Users Group is now established! They publish a most informative 36-page periodical called LOGon, which contains articles, reviews, listings, and other interesting information, all in French. For more information, write Association Francaise des Utilisateurs de Logo, 12 rue de la Montagne Ste Genevieve, Paris, France 75005.

What If

by

Michael Tempel

and Marie Bourgoin

In March's NLXual Challenges column, Robs Muir presented our challenge of writing IF in Logo. Here are the first three versions we came up with at LCSI.

```
     TO MICHAEL'S.IF :CONDITION  :TRUE.LIST
:FALSE.LIST
     GO :CONDITION
     LABEL "TRUE RUN :TRUE.LIST STOP
     LABEL "FALSE RUN :FALSE.LIST
     END
```

Since this abomination uses GO and LABEL, we were determined to find something better.

```
     TO YVES'.IF :COND :TLIST :FLIST
     MAKE "TRUE :TLIST
     MAKE "FALSE :FLIST
     RUN THING :COND
     END
```

This was quickly converted into an elegant one liner.

```
     TO BRIAN'S.IF :COND :TRUE :FALSE
     RUN THING :COND
     END
```

These IFs work the same way as the Logo primitive, with two differences. They each require three inputs. Also, infix operations must be enclosed with parentheses.

```
   BRIAN'S.IF (2=2) [PRINT "YES][PRINT "NO]
   YES
   BRIAN'S.IF  MEMBERP "A [A E I O  U][PRINT
[WE HAVE HERE A VOWEL.]] []
   WE HAVE HERE A VOWEL.
   BRIAN'S.IF  MEMBERP "X [A E I O  U][PRINT
[WE HAVE HERE A VOWEL.]] []
```

In the last example above, nothing happens because we have used an empty list for the "false" consequence.

TRUE WORD, FALSE WORD

An interesting feature of Logo is revealed by these procedures. The Logo predicates (MEMBERP, EQUALP, =, and the like) output either TRUE or FALSE; but TRUE and FALSE are ordinary Logo words and can be manipulated as such.

```
     PRINT WORD (EQUALP "APPLE "IBM) (6>1)
     FALSETRUE

      PRINT SENTENCE "NIAGARA (WORD (BUTLAST
BUTLAST 1=2) "LS)
     NIAGARA FALLS

     MAKE "FALSE [THIS IS NOT TRUE.]
     PRINT THING (4=3)
     THIS IS NOT TRUE.
```

Next month, we will reveal the rest of our IFs and some of those sent in by NLX readers. Meanwhile, if you have an IF, send it to Michael Tempel, Logo Computer Systems Incorporated, 555 West 57th Street, Suite 1236, New York, NY 10019.

Both authors are employed by LCSI. Michael Tempel is the Director of Training and Mario Bourgoin is a Programmer / Analyst.

NLXual Challenges

by

Robs Muir

With only a few sterling exceptions, the current literature about the Logo family of languages is preoccupied with children's experiences. Nowhere is this more obvious than in the catalogs of "educational" suppliers, where scores of books highlight the language "designed for children." Indeed, the original intent of the Logo project at MIT was to create a computer language / environment that had "no threshold and no ceiling" suitable for younger children and "naive users" as well as sophisticated computer scientists. Current publications about Logo often choose to emphasize the simplicity of Logo to the exclusion of the more advanced aspects of the language.

Logo has powerful implications for "mature" computer users. Evidence for this can be found in such texts as <u>Turtle Geometry</u> by Abelson and diSessa, <u>Thinking About [TLC] Logo</u> by Allen et al., the MIT Logo project memos, and the body of literature about LISP - the petri dish for the germ of Logo. Despite these outstanding works, there is little evidence that Logo is being taken seriously in higher education, Tom Lough's editorial in the March NLX notwithstanding. There are few exceptions.

Dr. James Armentage, of the Claremont Graduate School, is currently using both Logo and Pascal as notational languages in his graduate level course in Information and Computer Science. The school's degree program is tailored to professionals already in

NLXual Challenges continued

the field of information management who are interested in doing postgraduate work. Jim's course, entitled "Data Structures," includes Pascal only as a concession to the strong inductry focus of some of the students. Jim has found Logo to provide a list processing notation ideally suited for exploration of comples concepts in data structure.

LOGO HOMEWORK CHALLENGE

This month's challenge is similar to one of the first assignments given to Jim's students. They are asked to go to the microcomputer lab, read the reference manuals on Logo, and begin programming. This is most likely their first exposure to Logo. Here is their assignment (and yours): Write a turtle graphics program that will represent a list of objects in a visual tree structure.

Allow me to illustrate...

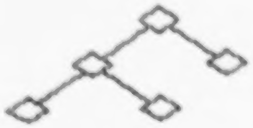
For the purposes of this challenge, let's select a list notation which has three elements: [node left.branch right.branch]. For example, the simple list [A B C] could be represented as

```
      A
     / \
    /   \
   B     C
```

A more complex list like [A [B C D] E] could be represented as

```
        A
       / \
      /   \
     B     E
    / \
   /   \
  C     D
```

With this quick background about lists, write a program that displays a list of non-primitive objects in this three-part format as a visual representation. You might choose to use little boxes to represent the list objects; this will be easier for the turtle to handle. It might also be possible to use the textscreen for this task, although this is beyond the scope of Armentage's original assignment.



Here are a couple of lists to use for testing your program.

[TALLER? BETTY [TALLER? TOM ROBS]]
[A [B C D] [E F [G H I]]]

In passing, you might notice that this process is similar to the format of prefix arithmetic operations. For example, compare the operation OUTPUT (PRODUCT 2 (SUM 3 4)) with the TALLER? list above.

Send your working solution (or even a valiant attempt) along with a self addressed stamped envelope to:

NLXual Challenges
Attn: Robs Muir
1688 Denver Avenue
Claremont, CA 91711

After compiling the contributed solutions, we will select several of the most interesting or unusual ones and redistribute them to each contributor.

Robs Muir is a teacher in the Claremont (CA) Unified School District and Claremont Graduate School.

The National Logo Exchange, copyright (c) 1985 by Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly, September through May, $25 (US) per year, mailed first class from Charlottesville, VA. $5 (US) additional per year for addresses outside the USA, Canada, and Mexico. The opinions expressed by the authors are not necessarily those of The National Logo Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress Street, Salem, MA 01971. Address other correspondence to: The National Logo Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor......................................Tom Lough

THE NATIONAL LOGO EXCHANGE

Volume 3 Number 9 FORWARD 100! May 1985

Too Many Questions, Not Enough Me

by
Leslie Thyberg

Instead of the more traditional hand raising or calling for "Help!" my students raise help flags as teacher call signals. During our Logo classes, help flags used to be up all over the place! I felt like a flight attendant answering call buttons!

Even though I have a team partner and other assistants, it was frustrating. There I was, trying to create a truly individualized child centered learning environment as a demonstration for a program of instruction called the Adaptive Learning Environments Model (ALEM). The learning environment was supposed to enable K-3 children to acquire basic academic skills while becoming more confident of their abilities to learn and cope with the social and academic demands of the classroom. Then Logo came to my classroom, and the forest of help flags sprang up.

More often than not, when I responded to a signal, the student would say, "I don't know what to do." My response, "What do the directions say to do?" typically resulted in "I don't know. I didn't read them." So much for idealism!

I had to come up with a mechanism to protect my sanity and still promote the independence in children I believe to be so important.

WHAT TO DO?

The solution to my dilemma came at the end of a particularly frustrating day. I was seated at my desk at home late in the evening, head in my hands, thinking about the "three" I's," things I had been trying to provide to my 65 students: Individualization, Interaction, and Integration. How could I help them with their questions and still encourage independence?

The words seemed to come from nowhere. But, as soon as I thought of them, I knew they were right.

"Ask three before me!"

I went dashing in to share the news with my husband!

The next day, the children learned the new rules! Before anyone asks me a question about Logo (or anything else), he or she must ask at least three other classmates. If no satisfactory answer is found, then the help flag can be used.

BLESSED RELIEF

How do you spell relief? T.H.R.E.E. It was fascinating to observe the effect of this rule on the children. To be sure, the noise level went up a bit. But now one could hear things like, "Did you ask Ellyn? She knows a lot about math." and "Yeah. I got stuck on that last week. Here's what I did..."

Children were automatically recognizing and promoting peer expertise and diminishing the stigma of making mistakes and failing when taking risks. And not a one of them had read Mindstorms!

There is something delightful about a five year old asking a third grader for assistance (and vice versa!). Affirmation, approval, and cooperative learning combined to make the classroom "our room." From trying a program, answering a question, asking for more details, the feedback made the peers coaches for each other.

As for me, I tried to be a model for this type of behavior, too. I encouraged students to plunge right in, and not be afraid to make mistakes. Pay attention to feedback, and try again. I do not know all the answers, but I am eager to help them discover a workable solution, if they are willing to use all the resources available to them.

Questions continued

A SPECIAL VISITOR

Our demonstration classroom has visitors in it all the time. It is the ultimate fishbowl! One day last spring, Prof. Seymour Papert was among the observers! He seemed quite interested as I described our "Ask three before me" rule to him.

Little did I know that he would cite it publicly at the Logo 84 conference at MIT later that summer! Since that time, I have received numerous requests for more information. How is it possible to create such a microworld?

My understanding of Papert's vision is that learning ought to be personal. The three I's allow for such personalization, especially when coupled with the "three before me" rule.

First, I arrange for students to select and sequence their own activities and materials, so their work is completely individualized. Next, I encourage interactions in which students teach each other and focus on each other's work. Third, I try to create an overall learning environment in which all subjects are integrated so thoroughly that students can effectively synthesize knowledge structures in their own minds. And it seems to work.

MORE THAN SURVIVAL

My aim is to continue providing opportunities for individualization, interaction, and integration in my classroom. My continuing challenge is to create and support an environment in which children, parents, and teachers can work with Logo cooperatively and creatively.

In that regard, "three before me" is more than a survival tool. It has become a methodology through which I have become a coach, counselor, consultant, collaborator, and friend to my student colleagues.

Teaching has become more challenging, more rewarding, and more exciting than ever as a result.

Just ask any three of my students!

Leslie Thyberg is a demonstration teacher / lecturer for the demonstration classroom of the Learning Research and Development Center at the University of Pittsburgh.

From the Editor

See More, Say More

As our columnists turned in their articles for this issue, they all echoed the same thoughts. "Is this really the last issue until September? I can't believe how the time has gone by!"

The passage of time causes change in us all. Logo can help make those changes positive ones, not just in programming, or how we solve problems, but in how we deal with others, and how we think about ourselves.

A few weeks ago, I received a brochure for an educational computing conference. In the section for Logo presentations, it mentioned "Seymore" Papert.

Not too long ago, my typical reaction would have been, "Hmmph. Doesn't anyone know how to spell? How could anyone make such a ridiculous mistake?" However, through Logo, I am learning the importance of accepting others and their work. I have made enough mistakes of my own to realize that none of us is anywhere near perfect.

When I saw "Seymore," it occurred to me that Seymour Papert has helped me to "see more" of the exciting world of learning and teaching than I had ever imagined possible.

Through Mindstorms and Logo, I can "see more" of myself, my students, and the work before us.

Indeed, I found that I rather like the spelling of Seymore.

Posy commented that it reminded her of "say more." Well, why not? Papert has always encouraged each of us to say more about the work we do with Logo, and about what we perceive to be the most effective methods of using this powerful learning tool.

See more. Say more. Sounds like two-thirds of some sort of a slogan. Do more? Sure. Let's resolve to do more for our students, for Logo, for ourselves. This summer, set aside some time to observe and to meditate, to argue and to mediate, to work and to cogitate.

See more. Say more. Do more. An agenda for summer Logo action.

To the many readers who are seeing the NLX for the first time, a hearty welcome! We hope you enjoy this issue. To our regular subscribers, see you in September! Thank you for your letters and your support this past year. I hope our paths cross at Logo 85 or WCCE 85.

FD 100! Tom

Volume 3 Number 9 The National Logo Exchange NLXTRA Supplement May 1985

NLXTRA

More What If

by Michael Tempel

In the March NLX, Robs Muir presented our challenge to write a Logo procedure which does the same thing as the IF primitive. (See NLXual Challenges, NLX March 1985.) Last month, we shared a few of our "solutions." (See "What If," NLX April 1985.)

A number of readers sent in their versions in response to the challenge. We were unable to digest them all fully in time for this issue of the NLX, so we'll defer a full discussion until later. For now, here are some more things to think about if you're trying to recreate Logo primitives in Logo.

IT'S FUNDAMENTAL

The exercise brings into focus a fundamental aspect of Logo; it is extensible. If something you need is not available as a Logo primitive, you can write a procedure of your own. Your procedure works generally the same way as a primitive, but this is not true for all situations. We mentioned two differences in the April article.

One difference is in the number of inputs. While most versions of Logo provide an IF that takes an optional third input, user defined procedures must have a fixed number of inputs.

A second difference between user defined procedures and primitives came to light when people tried to use infix operations with their IFs. For example,

```
IF 2 < 4 [PRINT "YES] [PRINT "NO]
```

works with the primitive IF, but not with IF procedures you and I might write in most versions of Logo. This is because parsing is not the same for user defined procedures as for primitives. In this example, Logo takes the 2 as the first input to an IF procedure and then runs into trouble. With the primitive IF, the expression 2 < 4 is first evaluated to be FALSE. This result is then used as the first input to IF.

Putting parentheses around the expression (2 < 4) solves the problem. (Ed. Note: Prefix notation with no parentheses worked with my Apple Logo. IF < 2 4 [PRINT etc.)

Another solution used by some people was to write procedures that expected the condition to be entered as a list. For example,

```
TO .IF :COND :TRUE :FALSE
RUN THING RUN :COND
END

.IF [2 < 4] [PRINT "YES][PRINT "NO]
```

The second RUN takes the :COND list as input and outputs the result of running it, in this case, the word FALSE.

Some newer Logos (Apple Logo II, Atari Logo, Coleco SmartLogo, and Logo for the Macintosh) are designed so that this problem does not arise. The user defined IF will accept inputs the same way as the primitive. In these Logos, the extensibility of the language is more fully realized because user defined procedures behave more like primitives.

THIRD WORD

There is a third, and probably more fundamental, discontinuity between user defined procedures and primitives in all versions of Logo. IF (and some other primitives, including RUN) may be either a command or an operation, depending on the input. User defined procedures must be one or the other, regardless of the input. Compare these two one line procedures:

```
TO FLIP1
IF  EQUALP 1 RANDOM 2 [OUTPUT "HEADS][OUTPUT
"TAILS]
END

TO FLIP2
OUTPUT IF EQUALP 1 RANDOM 2 ["HEADS]["TAILS]
END
```

In FLIP1, IF is a command, In FLIP2, IF is an operation. It is not possible to write a procedure which behaves like this. Or is there...?

Michael Tempel is the Director of Training for Logo Computer Systems Incorporated (LCSI).

Tail of a Turtle: the Valiant Effort

by

Glen Bull and Paula Cochran

In the beginning, there were turtles. Plexiglass hemispheres rolled across the floor, and delighted children of all ages as they drew elaborate designs with a pen in their belly. These floor turtles were the ancestors of modern screen turtles, triangles of light darting across the screen. Screen turtles soon left the laboratory, and can now be found in schools and homes everywhere. Their ancestors, the floor turtles, have been slower to evolve.

For primary and elementary children, a floor turtle may be more valuable than a printer. A floor turtle operates in the real world. When children play turtle, they can walk beside a real turtle. Combined with Tom Lough's NLX ABC's, the turtle can draw letters across a strip of paper to make a banner, for example.

A TURTLE, PRECISELY

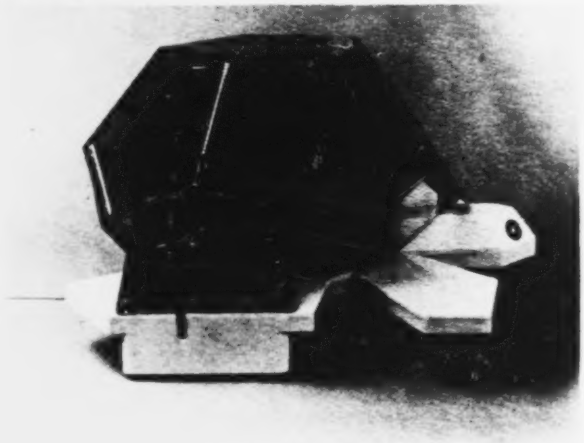
Recently we received a Valiant floor turtle from Harvard Associates (260 Beacon Street, Somerville, MA 02143 (617) 492-0660). The Valiant floor turtle is made in England. The examination model we received appears to be a precision instrument. When it draws a closed figure on a piece of paper on the floor, it returns to its original starting position. The current price of the Valiant turtle is less than most printers (about $400). This is the best combination of price and performance we have found in a floor turtle.

CUT THE CORD

One of the most remarkable features about the Valiant is the absence of a cable connecting the turtle to the computer. Instead, the computer talks to the turtle with an infrared light beam. (According to the manual, the range of the light beam is about 18 feet.)

At first, we made use of this feature by sending the turtle out into the hall. It was fun to watch the stares of startled passersby as the turtle scuttled under foot. However, there are more serious reasons for eliminating the cable. As young children use floor turtles connected to the computer, the cable gradually becomes tangled as the turtle turns round and round. Teachers are forced to devise ingenious strategies to deal with this problem. However, the best solution is the one used by Valiant: eliminate the cable altogether.

We were able to control the Valiant turtle with regular Logo commands for the versions we tested. (Some floor turtles require special commands instead of regular Logo commands, such as TFD in place of FD.) To make the Valiant turtle follow the same commands as the screen turtle, the user simply types: TURTLE ON. To turn the turtle off, the user types: TURTLE OFF. This is important because it allows a child to make use of procedures created before the turtle arrived.



IS IT ALIVE?

The Valiant floor turtle looks like a turtle. It has a head and feet, and a shell shape. Children can easily tell in which direction the turtle is heading.

One feature the Valiant turtle does not have yet is touch sensors. A touch sensor would require a two-way infrared link between the turtle and the computer, adding to the expense and complexity of the turtle. We prefer an affordable turtle without touch sensors.

Interfaces for the Valiant turtle are currently available for Apple, IBM, and Commodore 64 computers. The turtle comes with an infared box for transmitting commands, and can be connected to the computer through a variety of serial and parallel ports. The task of supporting all versions of Logo with all brands of input/output (I/O) ports available for computers is a massive task. Harvard Associates is currently developing software for additional versions of Logo, and for different kinds of I/O ports.

Valiant continued

The Valiant turtle appears to be a remarkable instrument for the price. We hesitate to use the word "break-through" until we can do more extensive testing, and see whether all models are as precise as the one we received for review. If it lives up to its early promise, the Valiant turtle may be the advance that makes turtles practical for kindergarten.

At one time computers in the classroom were a revolutionary idea. Now it is routine to see toddlers playing with them. Some day soon, floor turtles may be as common a sight as printers. And the Valiant turtle is pointing the way.

Glen Bull and Paula Cochran are regular columnists for the NLX.

ExperLogo: Bunny Power For the Macintosh

by Steve Tipps

As soon as the Apple Macintosh computer was introduced, the vision of a Logo version began to form in the heads of Logo lovers everywhere. The first such version is now available -- ExperLogo by ExperTelligence. I have had a demonstration copy of it for the last month and have been sneaking time to play with it. The time has been too short to fathom all it can do.

BUNNY FUN

One feature apparent from the demonstration disk is the phenomemonal graphics display which ExperLogo and the Macintosh can make. In this version, the graphics are performed by a triangular "bunny" which obeys all of the familiar turtle commands. I was a little bemused by changing animals in mid-Logo, but had no real problems with SHOWBUNNY instead of SHOWTURTLE.

ExperLogo comes with a regular 2-dimensional bunny, and a 3-dimensional bunny which can YAW, ROLL, and PITCH. The results of 3-D drawings are still unpredictable, but I did not know LEFT from RIGHT before I started with "regular" Logo. This one may take awhile longer to master, but I am enjoying it.

ExperLogo takes advantage of the windowing in Macintosh for graphics areas and different edit screens. The Listener Window acts like the immediate mode, while all procedures are written in edit windows. My personal QWERTY makes me a little uncomfortable with multiple windows. I found myself wishing for the simplicity of what I already knew.

MOUSING ABOUT

The mouse is yet another animal to deal with in the Macintosh ExperLogo environment. Programs which are primarily mouse implemented (such as MacDraw and MusicWorks) seem easy to use. Other Macintosh software is keyboard intensive (such as MacWrite). Programs which depend equally on mouse and keyboard seem cumbersome. ExperLogo is in this category. At the same time, I really liked being able to scroll up and down the Listener Window and activate any of commands I had written.

Other Macintosh features incorporated in ExperLogo include the Macintosh text fonts and flood-fill capabilities.

Additionally, you have the option of seeing the results of all your procedures by choosing SHOW OUTPUT. This is especially handy when debugging programs. You can see the picture and the values at the same time.

NOT SO FAST

ExperLogo is a compiled version which should make it extremely fast. Since I was working on a 128K Macintosh, I didn't benefit from the speed of the program compilation. In fact, on the 128K, the Macintosh accessed the disk drive on almost every line of code. Although the graphics were fast, the time to access the disk slowed work down a lot. I often found myself waiting not-so-patiently for execution.

I am going to keep on working with ExperLogo, and learn more of its special capabilities. Right now, I'll say I enjoyed working with Logo on the Macintosh. I like ExperLogo, but I don't know whether I love it. I applaud the effort made by ExperTelligence to bring Logo to a new audience. File handling, speed, and memory are all essential to moving Logo into other applications. ExperLogo has its heart in the right place, but my hands haven't quite mastered all its new and exciting possibilities.

ExperLogo has a list price of $149.95. For more information, write:

ExperTelligence
559 San Ysidro Road
Santa Barbara CA 93108
(805) 969-7874

Steve Tipps is a regular columnist for the National Logo Exchange newsletter.

If you are making presentations or conducting workshops on Logo, and would like some information flyers about the NLX for your attendees, send us the details, and we will be happy to provide you with a supply. NLX Flyers, Attn: Kaye Knight, PO Box 5341, Charlottesville, VA 22905.

Logo Computer Systems Inc. (LCSI) and Educational Spectrum announce "A Workshop for Experienced Logo People," to be held August 5 - 9 in New York City. Advanced Logo programming and applications in mathematics, science, language, and social studies will be guided by a distinguished faculty of experienced Logo instructors and lecturers. For more information, call Michael Tempel at (212) 765-4780.

Logo Information For Teachers (LIFT) offers memberships and a copy of their latest 30 page member directory for $10. For more information, write LIFT, PO Box 5396, Plymouth, MI 48170.

Radio Shack's Tandy Home Education System (THES) package includes a 64K Color Computer 2, two joysticks, a modem, a disk drive, and a copy of Color Logo. The new Super Logo will be available this summer with THES. Scheduled for release also are versions of Digital Research's DR Logo for the Radio Shack Model 4 computer, and the Tandy 2000 and 1000. For more information, contact Ron Moore, Director of Curriculum Sales, 1400 One Tandy Center, Ft. Worth, TX 76102.

The West Coast Logo Conference, scheduled for November 21 - 23, 1985, has extended the deadline of its call for research papers to June 15. Reports on completed Logo projects which address a specific educational topic or problem are invited. For more information, write to L. Polin, West Coast Logo Conference, Pepperdine University, 3415 Sepulveda Boulevard, Los Angeles, CA 90034.

McGraw-Hill announces the Second Annual Teacher Education Conference, to be held July 21-26 in Williamsburg, VA. The program includes lectures, small group discussions, and hands on activities in Logo as well as other topics. Graduate credit is available. For more information, write to Mary Jim Cunningham, Conference Coordinator, Webster Division, McGraw-Hill Book Company, 1221 Avenue of the Americas, New York, NY 10020, or call (800) 223-4180.

Have you been wondering if you can use your parallel printer with Apple Logo II? Thanks to Nicole Michaud of Logo Computer Systems Incorporated (LCSI), NLX readers can find out how.

Make sure your Apple Parallel Card is in slot # 1. Set the DIP switches. Most printers operate correctly with these settings.

Switch:	SW1	SW2	SW3	SW4	SW5	SW6	SW7
Setting:	OFF	OFF	OFF	ON	ON	OFF	OFF

Now boot Apple Logo II and type

```
EDITFILE "STARTUP
```

Now define the following procedures:

```
TO PARALLEL.CARD
.DEPOSIT 29694 96
PUTBYTES [234 234 234 234 234] 29656
PUTBYTES [44 193 193 48 251 141 144  192
96] 29736
END

TO PUTBYTES :LIST :ADDRESS
IF EMPTYP :LIST [STOP]
.DEPOSIT :ADDRESS FIRST :LIST
PUTBYTES BUTFIRST :LIST :ADDRESS + 1
END

MAKE "STARTUP [PARALLEL.CARD]
```

Press Open-Apple-A to save these procedures in the STARTUP disk file. The next time you boot Apple Logo II, it will be set up automatically to work with your Apple Parallel Card and your parallel printer.

Do you have a CompuServe account? If so, browse through the ERIC Logo Database! Type GO LGO and you will be able to search through files of article and report titles by various categories (grade level, subject, etc.). Information on how to order copies of the material is also available.

Additionally, the Young Peoples' Logo Association is scheduled to have services available on CompuServe soon. Be on the lookout for notices in early May.

"Using Logo to Teach Elementary and Middle School Geometry," a 3-semester hour graduate course, is scheduled for the Spartanburg campus of the University of South Carolina, June 10-25. The instructor is E. Carol Ellis. For more information, contact Dr. Judith Prince, Director of Graduate Regional Studies, USC-Spartanburg, Spartanburg, SC 29303.

Tipps for Teachers

by Steve Tipps

Survey Sense III

Information from surveys and other data collection can be reported in many ways. People talk about which answers were the most frequent or least frequent. Numbers or percentages are used for reporting how many people preferred different flavors of ice cream.

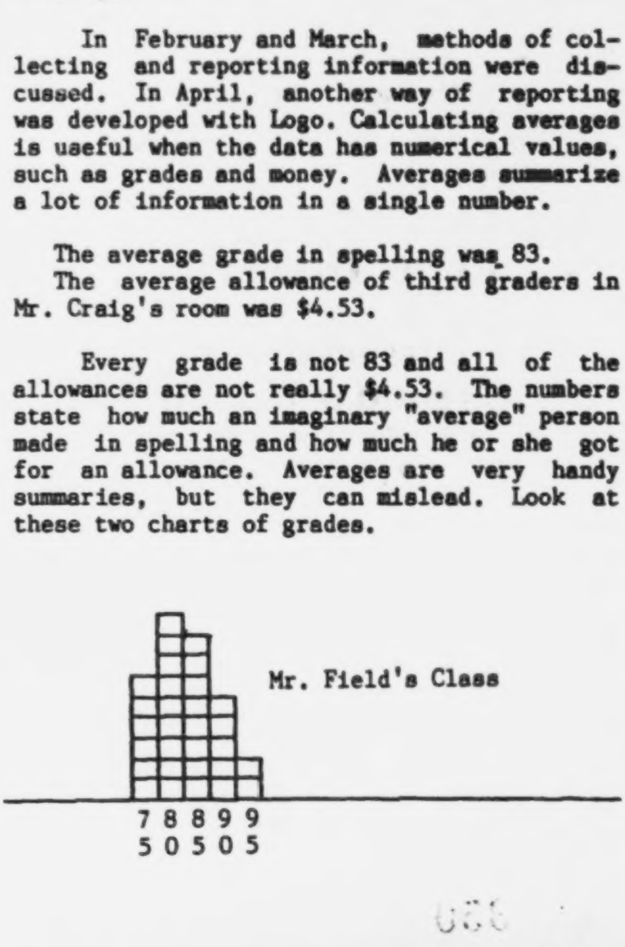
435 of 1000 people surveyed preferred vanilla ice cream.
43.5 percent preferred vanilla ice cream.

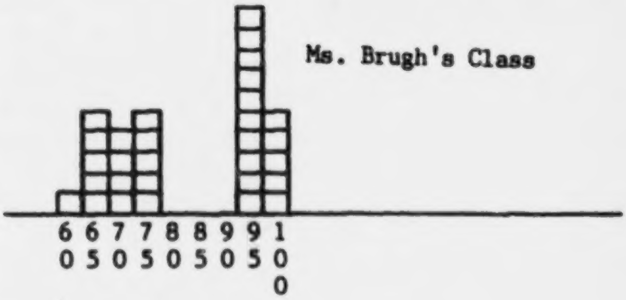
Only 10 of 1000 people liked licorice-strawberry bubblegum.
Only one percent liked licorice-strawberry bubblegum.

In February and March, methods of collecting and reporting information were discussed. In April, another way of reporting was developed with Logo. Calculating averages is useful when the data has numerical values, such as grades and money. Averages summarize a lot of information in a single number.

The average grade in spelling was 83.
The average allowance of third graders in Mr. Craig's room was $4.53.

Every grade is not 83 and all of the allowances are not really $4.53. The numbers state how much an imaginary "average" person made in spelling and how much he or she got for an allowance. Averages are very handy summaries, but they can mislead. Look at these two charts of grades.





Here are the two sets of grades.

```
MAKE "FIELD [75 75 80 80 80 85 85 85 90
95 80 85 80 85 80 85 90 90 75 75 80 85 80 85
75 75 90 90 95 80]

MAKE "BRUGH [65 65 70 70 75 95 95 95 100
100 65 65 65 70 75 95 95 95 100 100 60 100 95
95 95 75 70 75 75 95]
```

Both class averages are 83; but you can see that the scores in the two classes are not the same. The distributions do not look alike. Mr. Field's class has grades which are bunched around 80 or 85. Ms. Brugh's class has many high grades and a many low grades, with few in between.

HOW CLOSE TO AVERAGE?

Another calculation describes what you see on the chart. How much the scores are actually spread out is called the standard deviation. Simply put, the standard deviation tells how close the scores are to averge. Not so simply put, the standard deviation is the average of the sum of squares of the difference between each score and the average of all scores. You can get a full mathematical description of the standard deviation from any statistics textbook.

The Logo procedure S.D calculates the standard deviation. Since the AVERAGE is used as part of the calculation of the standard deviation, it is repeated from last month's column.

```
TO S.D :LIST
IF :LIST = [] OP SQRT :SUM.SQ / :NUMBER
MAKE "DIFF :AV - FIRST :LIST
MAKE "SUM.SQ :SUM.SQ + :DIFF * :DIFF
OP S.D BUTFIRST :LIST
END

TO AVERAGE :LIST
IF :LIST = [] OP :SUM/ :NUMBER
MAKE "SUM :SUM + FIRST :LIST
OP AVERAGE BUTFIRST :LIST
END
```

Tipps continued

Several variables need to be set up before using the procedures.

```
MAKE "SUM.SQ 0
MAKE "NUMBER COUNT :BRUGH
MAKE "SUM 0
MAKE "AV AVERAGE :BRUGH
PRINT :AV
83
PRINT S.D :BRUGH
14.1774
```

All of these steps can be put into one procedure to describe the list of scores.

```
    TO DESCRIBE :LIST
    MAKE "SUM
    MAKE "SUM.SQ 0
    MAKE "NUMBER COUNT :LIST
    MAKE "AV AVERAGE :LIST
    PRINT SE [THE AVERAGE IS] :AV
    PRINT  SE  [ THE STANDARD DEVIATION IS ]
S.D :LIST
    END
```

Now DESCRIBE the two lists of scores.

```
DESCRIBE :BRUGH
THE AVERAGE IS 83
THE STANDARD DEVIATION IS 14.1774

DESCRIBE :FIELD
THE AVERAGE IS 83
THE STANDARD DEVIATION IS 5.85946
```

The standard deviation for the narrow range of scores in Mr. Field's class is about 6, while the standard deviation for Ms. Brugh's class is much bigger. It is about 14 points.

In Mr. Field's class, most of the scores are within a range of 12 around the average. This is called plus or minus one standard deviation, or 83 + 6 and 83 - 6. The scores are much more spread out in Ms. Brugh's class. Most of the scores are within 14 above and below the average of 83.

In fact, the standard deviation states that about two-thirds of the scores are within the range described by the standard deviation. Count scores on the histogram to see if twenty of the scores in Mr. Field's class are above 79 and below 89. Then count to see if 20 of the scores in Ms. Brugh's class are between 69 and 97.

Your count is not perfect because we do not have bell-shaped, normal curves. However, you should be able to get a sense of how the standard deviation works. The average is important in summarizing where the center of the data is. The standard deviation tells whether the scores are bunched around the average or spread out.

NEXT YEAR WITH TFT

For the last three years, Tipps for Teachers has featured columns on many Logo commands and related activities. The topics have ranged from ones suitable for kindergarten teachers to those for high school mathematics classes. The range, flexibility, and potential of Logo is the sermon which I have been preaching monthly. That broad range of possibilities is central to NLX. Next year, this column will concentrate on Logo activities for grades 2 through grade 7 where most of the Logo activity seems to be. The emphasis will still be on Logo as a tool for exploring and understanding ideas, because integration of school subjects into Logo microworlds is crucial. I appreciate the many letters, comments, and suggestions which readers send. I hope that you will continue to challenge me as you challenge yourself with Logo.

Have an enjoyable summer!

Steve Tipps is the West Professor of Education at Midwestern State University in Wichita Falls, TX. He presents Logo workshops for school systems throughout the United States.

Logo Disserts:

Dissertations Dealing with Logo

by Barbara Elias

What factors and conditions influence the successful use of microcomputers by preservice and inservice teachers in a developing nation? This question was investigated by Dennis O. Harper in a computer laboratory set up at Kebangsaan University in Malaysia. Elementary and secondary teachers studied the Logo and PILOT languages in three pilot CAI courses. Randomly selected preservice secondary teachers who were already enrolled used PILOT to develop tests and tutorials in their subject area. Logo programming experiences for secondary volunteer teachers were designed to enhance basic math skills and problem solving abilities.

Disserts continued

Primary teachers who also volunteered for the study learned to use a special version of Atari Logo which permitted all of the programming to be done in their native language. These experiences emphasized programming and strategies for teaching and improving mathematics. Courses included instruction and laboratory time. The courses were conducted over a span of four months.

An analysis of pre- and posttest questionnaires and tests from two of the three groups indicated that Malaysian teachers:

(a) Expressed more positive attitudes toward computers and their use in education,

(b) Felt that computer use could be easily learned,

(c) Thought that computers were important for their country and would improve the quality of their lives, and

(d) Were more enthusiastic about computers than a group of American teachers who were given a similar attitude survey.

The results also suggest that English comprehension, mathematics skills and spatial abilities are significant predictive factors for success of the Malaysian primary teachers who use Logo. Harper feels that these results may well serve as a prerequisite for more serious study in computer assisted learning.

>>>Harper, Dennis O. (1984). Using Computer Assisted Learning for Teacher Training in Malaysia (Doctoral dissertation, University of California, Santa Barbara, 1984).

(Ed. note: Each month, Barbara Elias highlights a dissertation or thesis dealing with Logo. She is accepting copies of recent research results for the NLX Dissertation and Thesis Repository. Reports may be mailed to her in care of NLX, PO Box 5341, Charlottesville, VA 22905.)

Barbara Elias is an assistant professor in the Education Department of Virginia State University in Petersburg, VA, and is a doctoral candidate at the University of Virginia.

Interested in how to use Logo in your language arts classes? How about for science and math? Then come to Hawaii this summer for "Logo in the Classroom," taught by Glen Bull, Paula Cochran, and Tom Lough, July 8 - 19. A beginning Logo class will be taught by Elaine Blitman and Barbara Jamile. For more information, write Educational Spectrum, Inc., 21 East 40th Street, New York, NY 10016, or call (800) 223-2544. New York residents call (212) 686-5810 collect.

前 100

by
Hillel
Weintraub

If you are attending the Logo 85 conference this summer, don't miss Mr. Takito Totsuka's presentation on the use of Logo as a tool for developing awareness of natural phenomena. Totsuka-sensei ("sensei" is an honorific title given to teachers in Japan) himself is a powerful argument for both the use of Logo and the unimportance of Logo. Let me explain.

After learning about Logo, many teachers in Japan hesitate, offering comments such as, "This is really great, but I have no computers." or "I have only one computer for every twenty children." or "The Ministry of Education has control of the curriculum, and there is no place for Logo."

THE REAL POINT

These comments are missing the real point about Logo. Part of the fault lies with those of us who have become the Logo "spokespeople." We use Logo, we write and speak about Logo, and constantly extol its virtues. The focus often remains with Logo and computers rather than that of empowering the student through the learning process. I believe that this is the real point of Logo.

Let me tell you now about Totsuka-sensei. He and his students talk about stars and distances and shapes. They measure around the class. They watch the stars at night making their celestial drawings. They become more careful observers as a result of all their activities, one of which is Logo. Totsuka-sensei is an outstanding teacher who encourages his students to think, to look carefully, to question, to search for answers. Logo is a powerful tool for doing this. But, if his students didn't use Logo, they would still learn.

Papert's Mindstorms knocked me out with its message that we had a chance to affect the way computers were used in education. In particular, we could promote syntonic education. Unfortunately, we sometimes get mixed up into thinking that only through computers and Logo can we teach syntonically.

I'm trying to say that Logo on computers is powerful; discovery with or without computers is powerful; but, Logo and computers without discovery are just tools for enslavement instead of empowerment.

前 100 continued

VERY INTERESTING INDEED

This is an interesting time in the Japanese educational world. Some big changes are in the wind with the advent of computers into the classrooms. Committees of experts are already discussing things like what hardware will be used and who will develop software for students.

But I have not heard discussed the idea of students developing their own software, as is done in Logo explorations. That idea is as radical as having the students develop their own tests! Clearly a wider view of the future role of computers in education is needed.

I used to be worried that there wasn't enough use of computers in education in Japan. Now I'm frightened because the computers are coming, and the Logo idea (not the language) is an insufficient part of the plan.

I hope to see many NLX readers this summer at Logo 85 and WCCE 85. Until then, MAE 100!

Hillel Weintraub is a teacher at Doshisha International High School in Kyoto-fu. Japan, and is the president of the Society for Microcomputing in Life and Education (SMILE).

NLXionary

A Lectionary and Discussion of Logo Readings

by Griff Wigley

This month's readings are all in the January / February 1985 issue of the AEDS Monitor, published by the Association for Educational Data Systems. Enjoy.

"The Effects of a Logo Computer Programming Experience on Readiness for First Grade Creativity and Self Concept: A Pilot Study in Kindergarten," by Gordon Reimer.

>Pretests, twenty Logo sessions in twenty days, posttests, and voila! A treatment group of kindergarten students showed greater gains than a control group in ten of eleven basic skill readiness tests, two creativity measures, and in a self concept test. Statistically significant gains were made in the areas of visual discrimination, visual motor skills, visual memory, and number recognition. No mention is made of what the teacher actually did with either group of kids. The article includes a sizeable annotated summary of many of the positive claims made by Logo enthusiasts over the years.

"Four Instructor Roles in an Extended Logo Environment," by Andrew David.

>The author explores the notion of a microworld in the context of a question seldom asked, "What am I trying to do and how can Logo help me?" It is precisely the question not asked by Reimer in the research described above. (Instead, it was, "Let's do Logo because it has positive impact.") Educational research suggests all sorts of activities and strategies that a teacher can use day in and day out if the goal is to improve self concept, creativity, or basic skill readiness.

>Logo may not be the best choice. Given costs for teacher training and equipment, it may actually be a poor choice. David's final sentence is worth quoting here. "In seeking to redirect the educational process, we needn't look to any individual piece of software for the answers; rather, we must reflect on our lives, our experience, and our sense of what important work has to be done. Then we must decide what tools we need; we have added Logo to the many."

"The Implementation and Evaluation of a Logo Program for All Upper Elementary Teachers and Students in Three Schools," by Ann D. Thompson and Marilyn Blaustein.

>A description of a cooperative project between a university and a school district to implement Logo with 400 fourth, fifth, and sixth grade students in three schools. Again, no mention is made regarding the rationale behind the decision to implement Logo. No discussion on the delicate matter of deciding that all teachers would participate. No information on the reactions of the administration and school board to the original proposal, and their plans for ongoing support for the project. These are all extremely important details that the authors chose to ignore in their article, in favor of less helpful statistics such as what percent had attended previous workshops.

>The one almost helpful piece of information I received from the article is that the majority of participating teachers preferred to have the computer in their classrooms instead of in a central lab. But, alas, no information as to why is included.

NLXionary continued

"Haiku and Nim: Logo in the Language Arts," by Bates L. Hoffer and Pat Semmes.

>An interesting report of a project conducted in a university freshman writing course. The authors replicated a process similar to the NIM project that Abelson describes (see Apple Logo, p.103, or MIT Logo Memo 254). They began the project by giving the students Dan Watt's POET program, which generates a three line poem, and then instructed them to continually refine it until it generated realistic Japanese haiku. Some of the computer generated haiku were included in blind submissions to the university literary magazine and one was chosen for inclusion. This is a perfect example of using Logo as a tool to achieve a valid educational objective.

If you would like to comment on this NLXionary column or any of the previous ones, send your thoughts to me at 918 College Street, Northfield, MN 55057. Have a Logo summer!

Griff Wigley is a co-owner of Family Computing, Inc., a facilitator in the Faribault (MN) Public Schools, and a school board member of Prairie Creek Community School.

TurtleTips

by
Donna Lanyi
and Jane Toth

As the Logo season turns and another school year comes to a close, teachers and students look for ways to "show off" the work they have accomplished and the things they have learned.

TEACHER HINTS

A Logo computer fair is an excellent means of allowing students and teachers to "shine," keeping parents informed of Logo activities, and generating good public relations. Have students submit their procedures in various categories, including judged and non-judged.

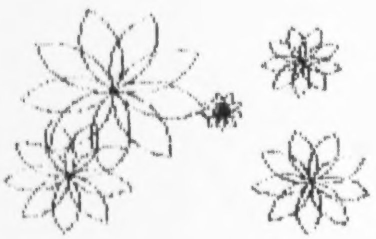
At the fair, arrange for several computers to be on hand so students can demonstrate their skills directly. Have student helpers assigned to encourage parents to try Logo themselves, and to assist them. Perhaps you could schedule mini-lessons for parents in advance.

At a recent ECCO (Educational Computer Consortium of Ohio) workshop we attended, we saw a new device you can use to make color displays of color graphics work, unless you are lucky enough to have a color printer or plotter. Kodak offers a special camera attachment called an Instagraphics CRT Slide Imager. This device and a CRT adapter combine together with a 35 millimeter single lens reflex camera to make instant color slides of color graphics on the computer screen. These slides can be projected in your display or used to produce color enlargements to give your fair a colorful accent.

The system is available through local Kodak dealers, but is rather expensive. The imager lists for around $340, the CRT adapter for about $40, and a special instagraphics color slide film pack with 10 exposures costs $12.95.

PROCEDURE OF THE MONTH

Spring has sprung! Enjoy our colorful flower garden to get you in the mood for the summer. Add stems and leaves, vary the patterns, and create bunches of bouquets for bulletin board displays or special gifts. Let the turtle become a honeybee, flitting from blossom to blossom.



For a listing of the procedure to draw this picture, send us a self addressed stamped envelope. We invite all elementary teachers and students to send their procedures and ideas to be shared. Donna Lanyi and Jane Toth, Beall Avenue School, 716 Beall Avenue, Wooster, OH 44691.

We would like to wish each of our readers a healthful summer. We look forward to hearing about your great Logo ideas for next year.

Donna Lanyi and Jane Toth use Logo in their elementary classrooms in the Wooster (OH) City Schools.

Listful Thinking

by

Glen Bull and Paula Cochran

Suffix Busters

Textbooks often talk about "exploring language". This phrase appears in the titles and chapter headings of many English texts. What does it really mean to explore language?

The word "explore" implies that you can be surprised. Students exploring the grounds of the University of Virginia will find a statue of Lewis and Clark. If Lewis and Clark had known what they would find when Thomas Jefferson sent them on their expedition, their trip would not have been as exciting.

It is hard to capture the excitement of exploration if you already know what is going to happen. A few rare individuals can play chess with themselves. They make a move from one side of the board, and then turn the board around and take the other side. For most of us, this kind of chess is a rather boring game. We can already see what the next move is going to be.

Pencil-and-paper language activities are sometimes like this. Although you are supposed to be "exploring" language, you can already see the next move coming. Exploration should leave room for a surprise.

This is where computers can supplement traditional pencil-and-paper activities. With properly constructed Logo tools, it is possible to capture and capitalize on the delight that comes from an unexpected result.

ENTERING THE MACHINE SHOP

In our April NLX Column, "Word Machines," we mentioned four pieces of Logo equipment for constructing tools for tinkering with words:

1) Separators
2) Joiners
3) Omitters
4) Testers

Joiners put pieces of words together to form new words. "Quick" and "ly" can be put together to form an adverb. Separators take words apart. "Eagerly" can be separated into "eager" and "ly". Testers can be used to check endings. If a word ends in "ly", it is an adverb ... or is it? One piece of equipment was not described last month: omitters. (Was it coincidence that omitters that were omitted?) What do you think omitters can do?

An omitter removes part of a word. The BUTLAST command is one of the simplest omitters. If there are two trucks and you want to remove one, BUTLAST is the command for the job.

```
PRINT BUTLAST "TRUCKS
TRUCK
```

The BUTLAST command prints everything BUT the LAST letter in a word. If you want to remove two letters, use two BUTLAST commands. What is the difference between a ship's captain and a ship's mast? Two letters, that's all.

```
PRINT BUTLAST BUTLAST "MASTER
MAST
```

CHANGING THINGS A BIT

A simple omitter can be used to change the structure of a variety of types of words. Its exact use depends on the needs of your class. The procedure CHANGE is an omitter that changes words by removing the last two letters. Here is what CHANGE looks like:

```
TO CHANGE :ITEM
OUTPUT BUTLAST BUTLAST :ITEM
END
```

You can CHANGE adverbs to adjectives,

```
PRINT CHANGE "QUICKLY
QUICK
```

nouns to verbs,

```
PRINT CHANGE "TEACHER
TEACH
```

and plural nouns to singular nouns.

```
PRINT CHANGE "BUSHES
BUSH
```

You can modify CHANGE so it will omit three letters, or one, by changing the number of BUTLAST commands in the procedure.

In a classroom, you might use a word machine like this as a jumping off point in a general unit about word suffixes, or a specific lesson about -es plural forms, root words for adverbs, and the like. In each case, the use of the machine is determined by the kind of words provided as "raw materials" by the teacher.

Listful Thinking continued

The CHANGE machine may be too predictable for older students. If your lesson requires that only certain suffixes be omitted, or that suffixes of differing lengths be omitted, who're you gonna call?

SUFFIX BUSTERS!

In order to create a Suffix Buster, we must first make a Suffix Detector by defining a list of suffixes all the same length.

```
TO TWO.LETTER.SUFFIXES
OUTPUT [ED ER ES LY OR]
END
```

Attention Terrapin and Commodore Users!

All procedures which follow are for LCSI Logos. The equivalent versions of procedures for Terrapin and Commodore Logo are found at the end of the column.

The Suffix Detector checks whether the last two letters of an item are in the list of two letter suffixes. To make this two-letter suffix detector, you will need the LAST.2.LETTERS procedure from last month. If you do not have this procedure, you will find a copy of it at the end of the column.

```
     TO TWO.LETTER.SUFFIX? :ITEM
     OUTPUT   MEMBERP  LAST.2.LETTERS  :ITEM
TWO.LETTER.SUFFIXES
     END
```

Now we can detect whether a suffix is in the list of suffixes used by the two-letter Suffix Detector.

```
PRINT TWO.LETTER.SUFFIX? "QUICKLY
TRUE

PRINT TWO.LETTER.SUFFIX? "BASEBALL
FALSE
```

You can change the way the two-letter Suffix Detector works by changing the list of suffixes in the TWO.LETTER.SUFFIXES procedure. For example, a list of two-letter suffixes consisting only of adjective endings could be created. Then the class could use the Suffix Detector to detect what type of suffix endings the detector was looking for. (For more suggestions on this sort of activity, see the All-Purpose Adjective Checker in last month's column.)

TO BUST

The Suffix Detector is the heart of a Suffix Buster. The abbreviation for BUTLAST is BL, and is used in the BUST procedure.

```
     TO BUST :ITEM
     IF THREE.LETTER.SUFFIX? :ITEM [OUTPUT BL
BL BL :ITEM]
     IF  TWO.LETTER.SUFFIX?  :ITEM [OUTPUT BL
BL :ITEM]
     IF ONE.LETTER.SUFFIX?  :ITEM [OUTPUT  BL
:ITEM]
     END
```

The Suffix Buster works on the suffixes you or your students have included in ONE.LETTER.SUFFIXES, TWO.LETTER.SUFFIXES, and THREE.LETTER.SUFFIXES. (Examples of these procedures are found at the end of the column.) Depending on the contents of these lists, the Suffix Buster can be used to omit suffixes of varying lengths in adverbs, nouns, adjectives, and verbs.

```
PRINT BUST "HAUNTED
HAUNT
```

The Suffix Buster is a stronger tool than the CHANGE machine because you have more control. For students using the tool, the working of the machine holds more surprise. In fact, if your students are doing a mystery exercise to determine what this machine does to words, the procedure name may be disguised as MYSTERY or MAGIC.

```
TO MAGIC :ITEM
OUTPUT BUST :ITEM
END

PRINT MAGIC "PLAYER
PLAY

PRINT MAGIC "BORROWS
BORROW

PRINT MAGIC "WONDERFUL
WONDER
```

Older students could be assigned the task of putting suffixes in the lists which will make certain changes for their fellow students. For example, can you put suffixes in the lists that will "capture" (and convert) only adjectives?

Listful Thinking continued

A OU EA I

Omitters needn't be confined to Suffix Detectors. For examples, it's fairly easy to detect and omit vowels. First create a list of vowels.

```
TO VOWELS
OUTPUT [A E I O U]
END
```

Use the list to make a Vowel Detector.

```
TO VOWEL? :LETTER
OUTPUT MEMBERP :LETTER VOWELS
END
```

The Vowel Detector works like this:

```
PRINT VOWEL? "A
TRUE

PRINT VOWEL? "B
FALSE
```

A Consonant Detector is almost as easy to make. Anything that is not a vowel must be a consonant.

```
TO CONSONANT? :LETTER
OUTPUT NOT MEMBERP :LETTER VOWELS
END
```

If you are thinking, "The letter Y is sometimes a vowel," we reply, "What a wonderful idea for a project! Can you make a Vowel Detector that also detects when Y is a vowel?"

Use the Consonant Detector to make a Vowel Omitter.

```
TO OMIT.VOWELS :ITEM
IF EMPTYP :ITEM [PRINT [] STOP]
TEST CONSONANT? FIRST :ITEM
IFTRUE [TYPE FIRST :ITEM]
OMIT.VOWELS BUTFIRST :ITEM
END
```

The Vowel Omitter only prints consonants.

```
OMIT.VOWELS "BASEBALL
BSBLL
```

In some ancient languages, such as Latin, the vowels were often omitted from inscriptions engraved in stone. (You might leave out a few letters too, if you were engraving words in stone, instead of using a word processor.)

Which carry more information in language - vowels or consonants? Claude Shannon developed the science of information theory to answer that type of question at Bell Laboratories. You can examine the same type of questions in your class by building a Consonant Omitter to go with the Vowel Omitter we provided.

The cryptic letters for the subheading of this section of the column are the remnants of a sentence which had the consonants omitted. Compare the same sentence with only the vowels omitted.

```
 a  ou ea  i     (consonants omitted)
Cn y   rd ths?   (vowels omitted)
```

THE YEAR IN REVIEW

In most schools, computers are still used primarily for science and mathematics. Although those are exciting uses, we think that computers should also be used for language arts. With Logo, almost everyone knows about graphics and pictures. List processing has the reputation of being an "advanced" use of the computer.

In schools across the land, teachers are being asked to use Logo in their classes. In many cases, teachers are teaching programming, because they don't know what else to do with Logo. We think that using Logo to teach may be even more important than teaching Logo.

This past year, we deliberately set out to create a column which would illustrate uses of Logo for playing with language. We hope that we have demonstrated that lists can be as much fun as turtling. We tried to describe ways of using Logo that allow the teacher to spend class time teaching English, not teaching Logo. If you found an activity that you could use with your class along the way, that was exactly what we intended.

One additional item: we now have an account on Special Net. If any of you would like to contact us, our identification is UVACOMDIS.

Glen Bull is a professor in the University of Virginia's School of Education, and teaches Logo courses at both the graduate and undergraduate level. Paula Cochran is a reformed English major who studied linguistics at the University of Cambridge. She is now a speech-language pathologist working with language-disabled children and Logo.

Listful Thinking Procedures

These procedures are the same for Apple, IBM, Terrapin, and Commodore Logo.

```
     TO ONE.LETTER.SUFFIXES
     OUTPUT [S]
     END

     TO TWO.LETTER.SUFFIXES
     OUTPUT [ED ER ES LY OR]
     END

     TO THREE.LETTER.SUFFIXES
     OUTPUT [IVE FUL ING]
     END

     TO LAST.2.LETTERS :ITEM
     OUTPUT   WORD  NEXT.TO.LAST  :ITEM  LAST
:ITEM
     END

     TO NEXT.TO.LAST :LETTER
     OUTPUT LAST BUTLAST :LETTER
     END

     TO LAST.3.LETTERS :ITEM
     OUTPUT  WORD  THIRD.FROM.LAST  :ITEM
LAST.2.LETTERS :ITEM
     END

     TO THIRD.FROM.LAST :LETTER
     OUTPUT LAST BUTLAST BUTLAST :LETTER
     END
```

The following procedures are for Apple and IBM Logo.

```
     TO BUST :ITEM
     IF THREE.LETTER.SUFFIX? :ITEM [OUTPUT BL
BL BL :ITEM]
     IF  TWO.LETTER.SUFFIX?  :ITEM [OUTPUT BL
BL :ITEM]
     IF ONE.LETTER.SUFFIX?  :ITEM [OUTPUT  BL
:ITEM]
     END

     TO ONE.LETTER.SUFFIX? :ITEM
     OUTPUT   MEMBERP   LAST   :ITEM
ONE.LETTER.SUFFIXES
     END

     TO TWO.LETTER.SUFFIX? :ITEM
     OUTPUT   MEMBERP  LAST.2.LETTERS  :ITEM
TWO.LETTER.SUFFIXES
     END

     TO THREE.LETTER.SUFFIX? :ITEM
     OUTPUT   MEMBERP  LAST.3.LETTERS  :ITEM
THREE.LETTER.SUFFIXES
     END
```

These procedures are for Terrapin Logo for the Apple and Commodore Logo.

```
     TO BUST :ITEM
     IF THREE.LETTER.SUFFIX?  :ITEM OUTPUT BL
BL BL :ITEM
     IF TWO.LETTER.SUFFIX? :ITEM OUTPUT BL BL
:ITEM
     IF  ONE.LETTER.SUFFIX?  :ITEM OUTPUT  BL
:ITEM
     END

     TO ONE.LETTER.SUFFIX? :ITEM
     OUTPUT   MEMBER?   LAST   :ITEM
ONE.LETTER.SUFFIXES
     END

     TO TWO.LETTER.SUFFIX? :ITEM
     OUTPUT  MEMBER?  LAST.2.LETTERS  :ITEM
TWO.LETTER.SUFFIXES
     END

     TO THREE.LETTER.SUFFIX? :ITEM
     OUTPUT  MEMBER?  LAST.3.LETTERS  :ITEM
THREE.LETTER.SUFFIXES
     END

     TO VOWELS
     OUTPUT [A E I O U]
     END

     TO VOWEL? :LETTER
     OUTPUT MEMBER? :LETTER VOWELS
     END

     TO CONSONANT? :LETTER
     OUTPUT NOT MEMBER? :LETTER VOWELS
     END

     TO OMIT.VOWELS :ITEM
     IF EMPTY? :ITEM PRINT [] STOP
     TEST CONSONANT? FIRST :ITEM
     IFTRUE PRINT1 FIRST :ITEM
     OMIT.VOWELS BUTFIRST :ITEM
     END
```

If you are using Version 1 of Terrapin Logo for the Apple, you will also need the MEMBER? and EMPTY? procedures below.

```
  TO MEMBER? :ITEM :LIST
  IF :LIST = [] OUTPUT "FALSE
  IF :LIST = "  OUTPUT "FALSE
  TEST :ITEM = FIRST :LIST
  IFTRUE OUTPUT "TRUE
  OUTPUT MEMBER? :ITEM BUTFIRST :LIST
  END

  TO EMPTY? :ITEM
  OUTPUT ANYOF :ITEM = [] :ITEM = "
  END
```

Teacher to Teacher

A Guest Review by Mark Lindsay

The New Teacher, Kids, and Logo

Teacher, Kids, and Logo and the new Turtle Trek Challenge Kits, by Carolyn Green and Christi Jaeger, make up a set of curriculum materials with a wide range of useful information for both beginning and experienced Logo teachers at the primary and elementary level.

Teacher, Kids, and Logo is a guidebook for a comprehensive Logo curriculum for kindergarten through sixth grade. I highly recommend it because it has an abundance of good Logo teaching ideas. Even the most dedicated advocate of discovery learning will find a wealth of information for use in the Logo classroom.

Turtle Trek Challenge Kits, a new feature, are now available for both the primary (K-3) and upper grade (4-6) levels. Each kit contains activity cards for 50 Logo projects.

The specific organization and contents of Teacher, Kids, and Logo were presented in an earlier review (NLX May 1984). One problem I found with the book was the authors' attempt to assign various Logo commands and concepts to specific grade levels. A nagging question haunts me. "What do I do when a child's explorations lead to a new concept or tool that may not be 'scheduled' for his or her grade level?"

The authors soften the guidelines by suggesting that "children's ability and creativity will take them in many directions and nothing precludes a command being introduced at any grade level." I feel this sentence should be highlighted and accompanied by examples of practice activities to lead children to a higher level of conceptualization.

WALK, DON'T RUN

Throughout the materials, the authors advise, "Walk, don't run through Logo!" This should be a command permanently programmed into every teaxcher's memory! Every time a teacher hurries a child, there is a price to pay.

This happened to me when I taught a group of fourth and fifth grade children in an after school class this past winter. The children were tested, using the authors' grade level objectives in the curriculum guidebook. All the children understood the concepts and commands through the fourth grade level. That was the successful part of the program.

The next step involved having the children choose project cards from the third and fourth grade selections of the Turtle Trek Challenge Kit. Based on the test scores, I assumed that they were ready for independent exploratory learning, with little introduction or discussion. WRONG!

I was running — not walking — the children through Logo! After fifteen minutes of chaos, I collected the project cards and selected just one, "Alien Music," to discuss.

We talked about how to set up a space scene with a UFO and how to integrate sounds and other space objects suggested in the example. Finally, we all began to relax. We spent the next four sessions setting up the project by using the suggestions on the activity cards and comparing our procedures to the "solutions" in the book. Because things did not turn out exactly like the model in the book, the children expressed feelings of inadequacy. So much for presenting models ... back to discovery learning!

A CLOSER LOOK

What happened? The wide range of activities on the cards is good and appear to be appropriately chosen for the two age levels. However, when implementing the activities with upper elementary children, I had difficulty understanding the directions on how to specifically reproduce the model.

For example, the "Alien Music" card suggested that a procedure could be written to draw a UFO, a star, or an asteroid. Perhaps some suggestions for simple models could be included. The student who needs more guidance could then build on the model and use more complex commands as they are learned. A "pure" discovery learning approach with few or no guidelines is frustrating for the novice. If guidelines are furnished, they could be ignored as sophistication in Logo is developed.

I feel the cards are an excellent addition to the classroom if they are used as an aid to learning Logo and not as THE way to teach Logo. They make great props to encourage problem solving, decision making, and creative thinking in elementary age children.

I found that it is helpful to jot comments on the backs of the cards after they have been used with the chidlren. These comments helped me remember what activities or specific directions caused difficulty and what the solutions were. Many successful future lessons could be built on the revisions suggested by these notes.

Teacher to Teacher continued

Teachers who use these materials need to define the concepts studied SLOWLY. We need to solve problems over a long period of time with children, and give them time to discover how to accomplish a task. The solutions book, included in the kit, should be kept on a high shelf, to reduce the temptation to use it as an "answer" book. This defeats the goal of the materials and the goal of Logo, too.

RUN, DON'T WALK

Run, don't walk, to get a copy of these excellent materials. Here is a listing of them all.

Teacher, Kids, and Logo book	18.95
Resource Disk and book	10.95
Turtle Trek Challenge Kit (K-3)	
-card set	16.95
-resource disk	10.95
Turtle Trek Challenge Kit (4-6)	
-card set	16.95
-resource disk	10.95
Poster and Command Chart Set	6.95
Edit Chart	3.50

These materials are available from Edu-Comp Publications, 14242 Wyeth Avenue, Irvine, Ca 92714.

YOUR TURN

Logo Information For Teachers (LIFT) is pleased to sponsor the Teacher to Teacher column in the NLX. We are interested in hearing from teachers who are integrating Logo materials effectively into curriculum areas. If you have the itch to author a Teacher to Teacher column as a guest reviewer, let us know. LIFT, PO Box 5396, Plymouth, MI 48170.

Mark Lindsay is a fifth grade teacher at Brietmeyer Elementary School in Detroit, MI, and an active LIFT member.

Logo 85, the Second International Logo Conference, is scheduled for July 22 - 25 at the Massachusetts Institute of Technology in Cambridge, MA. Themes announced for the conference include evaluation, teacher education, and the integration of Logo into the curriculum. Approximately 130 separate presentations are scheduled. Don't miss this exciting Logo event! For more information, write to the MIT Special Events Office, Room 7-111, Cambridge, MA 02139.

Q and A

by
Jim McCauley

Q. I have seen examples in Logo documentation of procedures that contain loops constructed with the primitives LABEL and GO. If loops can be created wtih tail recursion, why does Logo include these primitives?

<[-?-]>

A. The short answer is, "There is no useful reason to include LABEL and GO in a modern Logo that encourages good program design."

GO is directly analogous to GOTO in BASIC. LABEL is analogous to the line number used by GOTO. Here are parallel programs in BASIC and Logo that do the same thing.

```
100 REM **PRINT WORD N TIMES**
110 PRINT "WHAT WORD?"
120 INPUT WO$
130 PRINT "HOW MANY?"
140 INPUT HM
150 REM **BEGIN PRINT LOOP**
160 IF HM = 0 THEN GOTO 200
170 PRINT HM; " "; WO$
180 LET HM = HM - 1
190 GOTO 150
200 END
```

```
TO PRINT.N.TIMES
TYPE [WHAT WORD?]
MAKE "WORD FIRST READLIST
TYPE [HOW MANY?]
MAKE "HOW.MANY FIRST READLIST
LABEL "BEGIN
IF :HOW.MANY = 0 [STOP]
PRINT SE :HOW.MANY :WORD
MAKE "HOW.MANY :HOW.MANY - 1
GO "BEGIN
END
```

This ugly little exercise proves that it is perfectly possible to write BASIC programs in Logo. Here is a more elegant approach.

```
TO PRINT.N.TIMES
PRINT.LOOP GET.WORD [WHAT WORD?] GET.WORD
[HOW MANY?]
END
```

```
TO PRINT.LOOP :WORD :HOW.MANY
IF :HOW.MANY = 0 [STOP]
PRINT SE :HOW.MANY :WORD
PRINT.LOOP :WORD :HOW.MANY - 1
END
```

Q and A continued

```
TO GET.WORD :PROMPT
PRINT :PROMPT
OUTPUT FIRST READLIST
END
```

(There is no error checking, but it can be added easily, as we saw in the March Q and A column.)

LABEL and GO are Logo's very simple versions of PROG, an old fashioned construct in LISP (an artificial intelligence computer language after which Logo is modeled). Whe LISP was first created in the early 1960's, it was very restricted in its ability to handle iteration (looping). The early LISPs allowed simulation of loops through tail recursion, calling upon a function as the last step in that function's execution. While this produced something that resembled looping, it was enormously wasteful of computer resources, since the use of recursion created huge stacks of copied functions and data.

This was unsatisfactory, because looping is a very handy facility for many programming tasks, and most computer languages have good facilities for dealing with it. LISP developers added the PROG construct, the associated keyword GO and the SET or SETQ assignment statements (analogous to MAKE in Logo), all of which together made it easier to translate programming concepts from FORTRAN to LISP.

A significant number of LISP enthusiasts have always viewed this as a great leap backward for LISP, and they have been beavering away for years to eradicate this rather clumsy approach to iteration. There have been two alternatives.

1. Write a construct that is more elegant that PROG. A look at some microcomputer LISPs reflects some of the efforts of the larger LISP community. In muLISP, the LOOP construct simply places the entire iterative structure inside a list and executes it repeatedly until one or more conditions are satisfied. Other LISPs do similar things with the DO construct.

2. Teach people to define all loops as recursive structures, then make the LISP interpreter or compiler "smart" enough to figure out when a recursive call can be handled as though it were a loop. One of the LISPs that is cleverest in doing this is John Allen's TLC-Lisp. It can even handle some mutually recursive calls as if they were loops! This is a major accomplishment.

Although LABEL and GO are provided in most Logos, all Logos currently available for micros have some ability for handling tail recursion as iteration. See the November 1984 Q and A for more details.

I generally teach my students to build loops using tail recursion and give them names that include the word "LOOP", as I have shown above. In this way, the loop can be thought of as a "computing object" instead of being mixed in with a lot of other stuff in an overly long procedure. This makes life (and debugging) much easier.

(Ed. Note: If you have a question about Logo programming techniques or the thinking which leads to Logo programming, send it to NLX Q and A, PO Box 5341, Charlottesville, VA 22905. Each month, Jim McCauley answers selected questions in this column.)

Jim McCauley is a Coordinator of Computer Education for the Santa Clara (CA) County Office of Education, and has written Logo articles for many national publications.

YPLA Notes

The Young Peoples' Logo Association (YPLA) announces the Logo Liberty Challenge, an excellent class activity to encourage the use of Logo and help the Statue of Liberty at the same time. Have your students use Logo to show what Liberty means to them. Their procedures might draw pictures, print a poem, or tell a story. Send their procedures on a disk, along with a donation to the Statue of Liberty (a check or money order made out to "The Peoples' Campaign"), to the YPLA before October 15, 1985. You will be sent a disk of other Logo procedures in return. In addition, your donation and procedures will be forwarded to the Statue of Liberty project, and your procedures will be demonstrated at the West Coast Logo Conference, November 21 - 23, 1985, in Los Angeles.

The YPLA also announces their Logo Library II, a disk based update to their Logo Library. The disk is available for Apple II, Atari, Commodore 64, and IBM personal computers, and contains a wealth of Logo resource material. One especially helpful feature is a cross reference index of Logo commands and operations in a number of versions. Once you have purchased the disk, you may return it at any time for a free update. To order Logo Library II, send $29.95 ($24.00 for YPLA members) plus $1.05 shipping and handling ($31.00 total).

The YPLA has just published a 64-page book called <u>Logo Off the Computer</u>. It features stimulating and challenging activities which prepare students for Logo on the computer. Available now, the book costs $5.95 plus $1.25 shipping and handling.

For all offers above, send to YPLA, 1208 Hillsdale Drive, Richardson, TX 75081.

NLX Index

September 1984-May 1985

Volume 3 Number 1 September 1984

Logo and DOS: Classroom Partners, by Nancy Parks Sopp

From the Editor, by Tom Lough, Get Set

Tipps for Teachers, by Steve Tipps, A New Year with Logo

MAE 100 by Hillel Weintraub

Listful Thinking, by Glen Bull and Paula Cochran, Poetry Sparks

Apple Logo II: Its Bigger, Better, by Jason Gervich

Logo Disserts: Dissertations Dealing with Logo, by Barbara Elias

Q and A, by Jim McCauley

Teacher to Teacher, by Anne Cairns Federlein and Jim Fry, Logo Guided Discovery Kits

TurtleTips, by Donna Lanyi and Jane Toth

NLXionary of Logo Readings, by Griff Wigley

NLXual Challenges, by Robs Muir, It's about Time

September NLXTRA

Volume 3 Number 2 October 1984

Summer Fun with Logo, by Mary Bonness

From the Editor, by Tom Lough, Twenty/Twenty and More

Tipps for Teachers, by Steve Tipps, Signs of Logo

NLXionary of Logo Readings, by Griff Wigley

TurtleTips, by Donna Lanyi and Jane Toth

Listful Thinking, by Glen Bull and Paula Cochran, Pick-a-Treat

MAE 100, by Hillel Weintraub

Teacher to Teacher, by Regina Bowden and Anne Cairns Federlein, Triscuits and Task Cards: Aids for Elementary Grades

Logo Disserts: Dissertations Dealing with Logo, by Barbara Elias

Q and A, by Jim McCauley

NLXual Challenges, by Robs Muir, Random Rings and Things

Volume 3 Number 3 November 1984

Logo Trail of Tears, by C. M. Chmielowski

From the Editor, by Tom Lough, Fighting the Wheel

Tipps for Teachers, by Steve Tipps, Number High, Number Low

NLXionary of Logo Readings, by Griff Wigley

TurtleTips, by Donna Lanyi and Jane Toth

Listful Thinking, by Glen Bull and Paula Cochran, Count Your Blessings

Teacher to Teacher, by Jim Fry, Logo Exploration: An Adventure for Middle School Students

Q and A, by Jim McCauley

MAE 100, by Hillel Weintraub

Logo Disserts: Dissertations Dealing with Logo, by Barbara Elias

Turtle Song, by Tom Lough

NLXual Challenges, by Robs Muir, An XLent Problem

November NLXTRA

Volume 3 Number 4 December 1984

Functions of OUTPUT, by Nancy Kovatch

From the Editor, by Tom Lough, Looking and Seeing

Tipps for Teachers, by Steve Tipps, Numbers in Between

NLXionary of Logo Readings, by Griff Wigley

TurtleTips, by Donna Lanyi and Jane Toth

Really Random, by Patrick Salsich

Listful Thinking, by Glen Bull and Paula Cochran, Making a List and Checking It Twice

Teacher to Teacher, by Lou Wichert, Logo, Math, and Art Books for Grade 4 through 12

Q and A, by Jim McCauley

NLXual Challenges, by Robs Muir, Of Sunshine and Shadow

MAE 100, by Hillel Weintraub

December NLXTRA

Volume 3 Number 5 January 1985

Circles, Arcs, and Headaches, by Sharon Burrowes

From the Editor, by Tom Lough, Be It Resolved

Tipps for Teachers, by Steve Tipps, Wonderful Wiggles

NLXionary of Logo Readings, by Griff Wigley

TurtleTips, by Donna Lanyi and Jane Toth

Logo Disserts: Dissertations Dealing with Logo, by Barbara Elias

Listful Thinking, by Glen Bull and Paula Cochran, Adding Delete to your Tool Kit

Teacher to Teacher, by Anne Cairns Federlein, Guides for Teaching Logo to Young Children

Q and A, by Jim McCauley

NLXual Challenges, by Robs Muir, This One's for Cary

MAE 100, by Hillel Weintraub

January NLXTRA

Volume 3 Number 6 February 1985

Wordles and Turtles, by Cathy Frank

From the Editor, by Tom Lough, Stirring the Pot

Tipps for Teachers, by Steve Tipps, Survey of Surveys

NLXionary of Logo Readings, by Robbie and Griff Wigley

TurtleTips, by Donna Lanyi and Jane Toth

Listful Thinking, by Glen Bull and Paula Cochran, Power Tools

Logo Disserts: Dissertations Dealing with Logo, by Barbara Elias

Teacher to Teacher, by James Fry, Logo, Words, and Ideas: Teacher Training Materials

Q and A, by Jim McCauley

NLXual Challenges, by Robs Muir, Stand in a Corner

Volume 3 Number 7 March 1985

Up the Logo Ramp to List Operations, by Tony Stavely

From the Editor, by Tom Lough, Logo on the Move

Tipps for Teachers, by Steve Tipps, Survey Sense

NLXionary of Logo Readings, by Griff Wigley

TurtleTips, by Donna Lanyi and Jane Toth

Listful Thinking, by Glen Bull and Paula Cochran, Powerful Selections

Logo Disserts: Dissertations Dealing with Logo, by Barbara Elias

Teacher to Teacher, by Anne Cairns Federlein, Teacher Training Materials with Classroom Applications

MAE 100, by Hillel Weintraub

Q and A, by Jim McCauley

NLXual Challenges, by Robs Muir, What? IF?

March NLXTRA

Volume 3 Number 8 April 1985

The Pause That Debugs, by Harold Nelson

From the Editor, by Tom Lough, Logo and Sex

Tipps for Teachers, by Steve Tipps, Survey Sense II

NLXionary of Logo Readings, by Griff Wigley

TurtleTips, by Donna Lanyi and Jane Toth

Listful Thinking, by Glen Bull and Paula Cochran, Word Machines

Logo Disserts: Dissertations Dealing with Logo, by Barbara Elias

Teacher to Teacher, by Jan Cardwell, The Turtle Goes to Kindergarten

MAE 100, by Hillel Weintraub

Q and A, by Jim McCauley

What If, by Michael Tempel and Mario Bourgoin

NLXual Challenges, by Robs Muir

April NLXTRA

Volume 3 Number 9 May 1985

Too Many Questions, Not Enough Me, by Leslie Thyberg

From the Editor, by Tom Lough, See More, Say More

Tipps for Teachers, by Steve Tipps, Survey Sense III

ExperLogo: Bunny Power for the Macintosh, by Steve Tipps

NLXionary of Logo Readings, by Griff Wigley

TurtleTips, by Donna Lanyi and Jane Toth

Listful Thinking, by Glen Bull and Paula Cochran, Suffix Busters

Tail of a Turtle: The Valiant Effort, by Glen Bull and Paula Cochran

Logo Disserts: Dissertations Dealing with Logo, by Barbara Elias

Teacher to Teacher, by Mark Lindsay, The New Teacher, Kids, and Logo

MAE 100, by Hillel Weintraub

Q and A, by Jim McCauley

More What If, by Michael Tempel

NLXual Challenges, by Robs Muir

May NLXTRA

NLXual Challenges

by
Robs Muir

FORWARD, Ever FORWARD

The Logo turtle is a remarkably powerful object for creating elementary computer graphics. The power of Logo's turtle is especially obvious after coming from other programming environments -- or after trying to teach shape tables to fourth graders.

The simplicity of guiding a computational object such as the turtle around the screen with FORWARDs and RIGHTs obscures the underlying complexity of computer graphics. This is both good and bad. On one hand, the turtle frees us from the tedium of low level programming and allows us to explore larger areas, such as the Total Turtle Trip Theorem. This is what high level programming languages do best. The disadvantage with the turtle's simplicity is that it insulates us from some primary computer functions.

DOT'S ALL RIGHT

Using cartesian coordinates, we could create a DOT command that would require an input and draw a dot at a specified location, for example, DOT [50 50], or DOT 50 50. Indeed, some versions of Logo have DOT as a primitive. It might be clearer to think of "drawing a dot" as illuminating a "pixel" (picture element).

Most current computers create a "dot" on the screen by "turning on" a memory location somewhere in computer memory. The Logo commands, PENDOWN FD 1 BK 1, will cause the computer's "bit-mapped graphics page" to change its state slightly. However, most computers con't use the coordinate scheme that Logo uses to locate positions on its "bit map." Instead, screen coordinates (Cartesian coordinates) are used.

Logo translates the Cartesian coordinates it uses into screen coordinates that the computer uses to display a pixel. You might find it worthwhile to write a Logo procedure that translates from one coordinate system to another.

HERE IT COMES

This month's challenge is for all you people who use the turtle. I hope that doesn't exclude anyone! Given the previous discussion of coordinates, write a procedure that duplicates Logo's FORWARD command without resorting to FD or BK. Of course, you can't use FORWARD (a primitive name) so let's call this new procedure(s) MY.FD. The new MY.FD should function exactly like FD, that is, RT 45 MY.FD 50 RT 34 MY.FD 12 should create the same pattern as RT 45 FD 50 RT 34 FD 12.

The entire problem can be coded in Logo, so don't spend too much time looking at the Technical Reference Manual for your computer. This is the real beauty of expressive high level languages like Logo; you can simulate lower level behaviors using simple elegant syntax.

Send your working solution (or even a valiant attempt) along with a self addressed stamped envelope to:

NLXual Challenges
Attn: Robs Muir
1688 Denver Avenue
Claremont, CA 91711

After compiling the contributed solutions, we will select the most interesting or unusual solutions and redistribute them to each contributor.

MY.FD 100!

Robs Muir is a teacher in the Claremont (CA) Unified School District and Claremont Graduate School.

The National Logo Exchange, copyright (c) 1985 by Posy Publications, a part of The Posy Collection, all rights reserved. Published monthly, September through May, $25 (US) per year, mailed first class from Charlottesville, VA. $5 (US) additional per year for addresses outside the USA, Canada, and Mexico. The opinions expressed by the authors are not necessarily those of The National Logo Exchange. Permission is granted for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy articles herein for the flat fee of $2 per copy of each article. Payment should be sent directly to CCC, 21 Congress Street, Salem, MA 01971. Address other correspondence to: The National Logo Exchange, Box 5341, Charlottesville, VA 22905. ISSN 0734-1717

Editor..Tom Lough

Topical Index of the National Logo Exchange newsletter, Volumes 1, 2, and 3, September 1982 through May 1985.

Compiled by Regina Sapona.

The NLX Topical Index consists of five parts:

A. Author Index
B. Book Review Index
C. Product and Information Index
D. Column Index
E. Subject Index

Articles are located by the month and year of the NLX issue in which the article appears, followed by a colon and the page number of that issue. For example, Nov 82: 1 denotes the November 1982 NLX issue, page 1. Articles appearing in the NLXTRA (an irregularly published NLX supplement) are indicated by an X immediately after the page number. For example, Dec 84: 1X denotes the December 1984 issue, page 1 of the NLXTRA.

A. Author Index

Blitman, E. Nov 82: 1
Bonness, M. Oct 84: 1
Bourgoin, M. Apr 85: 15
Bowden, R. Oct 84: 10
Bull, G. Sep 82: 5; Oct 82: 5; Nov 82: 5; Dec 82: 5; Jan 83: 5; Feb 83: 5; Apr 83: 5; May 83: 5; Sep 83: 6; Oct: 83: 6; Nov 83: 6; Dec 83: 3,10; Jan 84: 4,11; Feb 84: 9; Mar 84: 9; Apr 84: 9; May 84: 1, 7; Sep 84: 7; Oct 84: 7; Nov 84: 7; Dec 84: 9; Jan 85: 9; Feb 85: 9; Mar 85: 8; Apr 85: 8; May 85: 1,8, 1X
Burke, M. Oct 83: 10
Burrowes, S. Apr 83: 1; Jan 84: 1; Jan 85: 1
Cardwell, J. Nov 83: 1; Apr 85: 12
Chmielowski, C. M. Nov 84: 1
Cochran, P. Sep 84: 7; Oct 84: 7; Nov 84: 7; Dec 84: 9; Jan 85: 9; Feb 85: 9; Mar 85: 8; Apr 85: 8; May 85: 8, 1X
Elias, B. Sep 84: 10; Oct 84: 12; Nov 84: 14; Dec 84: 1X; Jan 85: 8; Feb 85: 12; Mar 85: 11; Apr 85: 11; May 85: 4
Federlein, A. C. Sep 84: 12; Oct 84: 10; Jan 85: 12; Mar 85: 12
Frank, C. Apr 84: 1; Feb 85: 1
Fry, J. Mar 83: 1; Apr 83: 1; Dec 83: 5; Sep 84: 12; Nov 84: 10; Feb 85: 13
Gervich, J. Sep 84: 9
Gervich, S. Sep 83: 1
Kovatch, N. Dec 84: 1
Lanyi, D. Sep 84: 13; Oct 84: 6; Nov 84: 6; Dec 84: 7; Jan 85: 7; Feb 85: 8; Mar 85: 7; Apr 85: 7; May 85: 7
Lindsay, M. May 85: 12
Lough, T. Sep 82: 1, 2, 4, 6, 7; Oct 82: 1. 2. 7; Nov 82: 1, 2, 7;

Dec 82: 1, 2, 7; Jan 83: 1, 2, 7; Feb 83: 2; Mar 83: 1, 2; Apr 83: 2; May 83: 2, 10, 11; Sep 83: 2, 7; Oct 83: 2; Nov 83: 2; Jan 84: 2; Feb 84: 3; Mar 84: 3; Apr 84: 2, 14; May 84: 2, 11; Sep 84: 2; Oct 84: 2; Nov 84: 2, 15; Dec 84: 3; Jan 85: 2; Feb 85: 3; Mar 85: 2; Apr 85: 3; May 85: 2
Mangiapane, S. May 84: 13
Mass, L. Dec 83: 1
McCauley, J. Sep 84: 11; Oct 84: 13; Nov 84: 11; Dec 84: 14; Jan 85: 14; Feb 85: 14; Mar 85: 14; Apr 85: 14; May 85: 13
Muir, R. Nov 83: 11; Dec 83: 15; Jan 84: 15; Feb 84: 16; Mar 84: 15 Apr 84: 16; May 84: 15; Sep 84: 16; Oct 84: 15; Nov 84: 16; Dec 84: 15; Jan 85: 15; Feb 85: 15; Mar 85: 15; Apr 85: 15; May 85: 15
Nelson, H. Apr 85: 1
Nix, L. Sep 82: 1; Feb 83: 1
Pearce, E. May 83: 9
Rainey, J. Feb 84: 1
Razzano, L. May 83: 1
Reilly, P. May 83: 1
Rifkin, B. Jan 83: 1
Riordan, T. Mar 83: 5; Apr 83: 7
Salsich, P. Dec 84: 8
Sheffer, N. Mar 84: 1; Apr 84: 8
Sopp, N. Oct 83: 1; Sep 84: 1
Stavely, T. Mar 85: 1
Stevens, P. Feb 84: 4
Sugarman, J. Dec 82: 1
Tempel, M. Apr 85: 15; May 85: 1X
Tepper, L. Mar 84: 1
Thyberg, L. May 85: 1
Tipps, S. Sep 82: 2; Oct 82: 3; Nov 82: 3; Dec 82: 3; Jan 83: 3; Feb 83: 3; Mar 83: 3; Apr 83: 3; May 83: 3; Sep 83: 3; Oct 83: 3; Nov 83: 3,9; Dec 83: 2,3,6; Jan 84: 4,7; Feb 84: 5; Mar 84: 5; Apr 84: 3; May 84: 3; Sep 84: 3; Oct 84: 3; Nov 84: 3; Dec 84: 4; Jan 85: 3; Feb 85: 3; Mar 85: 3; Apr 85: 3; May 85: 3, 1X
Toth, J. Sep 84: 13; Oct 84: 6; Nov 84: 6; Dec 84: 7; Jan 85: 7; Feb 85: 8; Mar 85: 7; Apr 85: 7; May 85: 7
Towberman, S. Oct 82: 1
Weintraub, H. Apr 84: 5; Sep 84: 6; Oct 84: 9; Nov 84: 13; Dec 84: 1X; Jan 85: 1X; Feb 85: 1X; Mar 85: 13; Apr 85: 13; May 85: 5
Wichert, L. Dec 84: 12
Wigley, G. Mar 84: 3; Apr 84: 6, 13; May 84: 6; Sep 84: 13; Oct 84: 5; Nov 84: 5; Dec 84: 6; Jan 85: 6; Feb 85: 6; Mar 85: 6; Apr 85: 6; May 85: 6
Wigley, R. Feb 85: 6
Yee, D. Jan 84: 14
Young, K. May 84: 1

B. Book Review Index {also includes books mentioned in columns} [first author listed]
* see also "Teacher to Teacher," Volume 3 for reviews of additional materials

Abelson, *Apple Logo* and *Logo for the Apple*-- Dec 82: 1; Jan 83: 8
Abelson, *Turtle Geometry*-- May 83: 10
Allen, *Thinking about [TLC] Logo*-- Mar 84: 7
Bearden, *1,2,3, My Computer and Me*-- May 83: 10; Feb 84: 5
Bearden, *The Turtle Goes to Kindergarten*-- Apr 85: 12
Bitter, *Apple Logo Primer*-- May 83: 10
Bowen, *Apple Logo Training Manual*-- May 83: 10
Burnett, *Logo: An Introduction*-- Mar 83: 1; Nov 83: 5
Davis, *Logo: A Problem-Solving Approach*-- May 83: 10
Green, *Teacher, Kids, and Logo*-- May 84: 13; May 85: 12
Horn, *Apple Logo in the Classroom* [from MECC]-- May 83: 10
Horn, *ComLit, Computer Literacy for Kids*-- Jan 85: 12
Horn, *Introduction to Logo for Teachers Training Materials* [from MECC]-- Mar 85: 12
Mass, *Kids Working with Computers: An Apple Logo Manual for Turtle Graphics*-- Dec 83: 1
McDougall, *Learning Logo on the Apple II*-- Apr 84: 15
Moscovitch, *Explorer's Guide to Apple Logo*-- Nov 84: 10
Muller, *Turtle's Sourcebook*-- Jan 83: 1
Papert, *Mindstorms*-- Sep 82: 1
Nevile, *Let's Talk Apple Turtle*-- Apr 84: 15
Thornburg, *Computer Art and Animation: A User's Guide to Commodore 64 Logo*-- Dec 84: 12
Thornburg, *Discovering Apple Logo: An Invitation to the Art and Pattern of Nature*-- Dec 84: 12
Watt, *Learning with Logo*-- Jan 83: 8; Oct 83: 9

C. Product and Information Index

Apple Logo II--Sep 84: 9
Apple Sprite Logo-- Apr 84: 6
bulletin board (electronic)
    University of Virginia-- Jan 85: 11
    YPLA-- Sep 84: 1X
conference notes
    Closing the Gap--Jan 85: 15
    Logo 84-- Sep 84: 14; Oct 84: 6; Nov 84: 5; Dec 84: 6
Dissertation Repository--May 84: 15
ExperLogo [Macintosh]-- May 85: 3X
graphics dump-- Feb 85: 12
Infoword Logo-- Feb 84: 2
Kaypro Logo-- Sep 84: 16
keyboard templates-- Dec 84: 9
LCSI News (materials, toolkits)-- May 83: 10; Nov 83: 12
Local Logo Exchanges (LLX)
    Chicago-- Dec 83: 10
    starting an LLX-- Oct 83: 12
Logo around the world

  Australia-- Jan 84: 3; Sep 84: 1X
  Boston-- Mar 84: 14
  Japan-- Jan 84: 14 (see also Weintraub columns)
  Malaysia-- May 85: 4
  New Mexico-- Oct 83: 8
  Netherlands-- Mar 84: 14; Sep 84: 1X
  South America-- Sep 84: 1X
Logo Depository (materials for teaching)--Jan 85: 2
Logo for the Sony-- Apr 84: 5
Logo Information for Teachers (LIFT)-- Oct 84: 16; Apr 85: 2X
Logo in magazines (references)-- Feb 84: 14, 15; Mar 84: 8
Logo on sale-- Jan 84: 10; Feb 84: 4; Sep 84: 3
Logo notes and news
  Mar 83: 7 New Logo versions announced
  May 83: 11 Friends of LISP/Logo & Kids [FOLLK]
  Sep 83: 5 UNICOM has version of Terrapin Logo which will run on a Corvus hard disk networks;
  Oct 83: 9 Dagobert & Lissa Soergel have MIT/LCSI Logo comparison of commands
  Jan 84: 3 MECC announces new Logo products.
  Jan 84: 16 Quadlink board; Micro a new magazine from Japan
  Sep 84: 3 TV Turtle, weekly show from Canada
  Oct 84: 14 WIZ-Kit utility diskette, videotape
  Feb 85: 6 Terrapin utilities announced
  Mar 85: 13 PC Logo, Valiant Turtle, TLC-Logo
  Mar 85: 12 LCSI Sprite Logo, Terrapin network version
  Apr 85: 10 Logo versions, other products
  Apr 85: 11 upgrade for Terrapin, ExperLogo for Macintosh
Logo Penpal Network-- Sep 83: 8; Nov 84: 7; Sep 84: 2X
Logo publications (and other related newsletters)--
  ERIC anounces computer newsletter. Oct 83: 11
  Logo Some Z's-- Nov 84: 16
  Microquests-- Feb 85: 8; Apr 85: 13
  Pascal newsletter-- Feb 85: 2
  TurtleTalk-- Dec 83: 16
Logo survey results-- Sep 84: 10
News from LCSI--
  Nov 83: 12 materials for courses and workshops
    public domain disks-- Apple Logo Sample Program Disk, the Apple Logo Tool Kit; Apple Logo Presentation manual
  Nov 84: 1X prices
NOVA information-- Nov 83: 2
plotters (Sweet P)-- Mar 84: 11; Sep 85: 9
research-- Oct 83: 5; Oct 84: 7
Research Directory-- Oct 83: 12; Nov 84: 11
Spanish Logo from TI available-- Mar 84: 16
Special Case Registry (NLX)-- May 84: 15
sprite boards for the Apple-- Jan 84: 15
teaching materials-- Oct 84: 14; Dec 84: 9; Jan 85: 2; Mar 85: 12; Apr 85: 6, 12; also see "Teacher to Teacher," Volume 3
Tasman turtle-- Mar 83: 7

Terrapin news (updated Logo for Apple computers)-- Feb 84: 3
turtle puppets (Folktails Turtle from Folkmanis)-- Jan 84: 13;
Sep 84: 2X
Valiant Turtle--Mar 85: 3; May 85: 2X
YPLA-- Oct 83: 3; May 85: 14

D. Column Index

From the Editor-- Tom Lough

Sep 82: 2; Oct 82: 2; Nov 82: 2; Dec 82: 2; Jan 83: 2; Feb 83: 2, Mar 83: 2; Apr 83: 2; May 83: 2; Sep 83: 2; Oct 83: 2; Nov 83: 2; Jan 84: 2; Feb 84: 3; Mar 83: 3; Apr 84: 2; May 84: 2; Sep 84: 2; Oct 84: 2; Nov 84: 2; Dec 84: 3; Jan 85: 2; Feb 85: 3; Mar 85: 2; Apr 85: 3; May 85: 2

Listful Thinking-- Glen Bull & Paula Cochran

Sep 84: 7; Oct 84: 7; Nov 84: 7; Dec 84: 9; Jan 85: 9; Feb 85: 9; Mar 85: 8; Apr 85: 8; May 85: 8

Logo Disserts: Dissertations dealing with Logo-- Barbara Elias

Sep 84: 10; Oct 84: 12; Nov 84: 14; Dec 84: 2X; Jan 85: 8; Feb 85: 12; Mar 85: 11; Apr 85: 11; May 85: 4

MAE 100 - Logo in Japan-- Hillel Weintraub

Apr 84: 5; Sep 84: 6; Oct 84: 9; Nov 84: 13; Dec 84: 1X; Jan 85: 1X; Feb 85: 1X; Mar 85: 13; Apr 85: 13; May 85: 5

Microworlds-- Glen Bull

Sep 82: 5; Oct 82: 5; Nov 82: 5; Dec 82: 5; Jan 83: 5; Feb 83: 5; Apr 83: 5; May 83: 5; Sep 83: 6; Oct 83: 6; Nov 83: 6; Dec 83: 10; Jan 84: 11; Feb 84: 9; Mar 84: 9; Apr 84: 9; May 84: 7

NLXionary: A Lectionary of Selected Logo Readings-- Griff Wigley

Mar 84: 3; Apr 84: 6, 13; May 84: 6; Sep 84: 13; Oct 84: 5; Nov 84: 5; Dec 84: 6; Jan 85: 6; Feb 85: 6; Mar 85: 6; Apr 85: 6; May 85: 6

NLXual Challenges-- Robs Muir

Nov 83: 11; Dec 83: 15; Jan 84: 15; Feb 84: 16; Mar 84: 15; Apr 84: 16; May 84: 15

Q and A-- Jim McCauley

Sep 84: 11; Oct 84: 13; Nov 84: 11; Dec 84: 14; Jan 85: 14; Feb 85: 14; Mar 85: 14; Apr 85: 14; May 85: 13

Teacher to Teacher-- reviews by Regina Bowden, Anne Cairns Federlein, Jan Cardwell, Jim Fry, Mark Lindsay, Lou Wichert

Sep 84: 12; Oct 84: 10; Nov 84: 10; Dec 84: 12; Jan 85: 12; Feb 85: 13; Mar 85: 12; Apr 85: 12; May 85: 12

Tipps for Teachers-- Steve Tipps

Sep 82: 2; Oct 82: 3; Nov 82: 3; Dec 82: 3; Jan 83: 3; Feb 83: 3; Mar 83: 3; Apr 83: 3; May 83: 3; Sep 83: 3; Oct 83: 3; Nov 83: 3; Dec 83: 3; Jan 84: 7; Feb 84: 5; Mar 84: 5; Apr 84: 3; May 84: 3; Sep 84: 3; Oct 84: 3; Nov 84: 3; Dec 84: 4; Jan 85: 3; Feb 85: 3; Mar 85: 3; Apr 85: 3; May 85: 3

TurtleTips-- Donna Lanyi & Jane Toth

Sep 84: 13; Oct 84: 6; Nov 84: 6; Dec 84: 7; Jan 85: 7; Feb 85: 8; Mar 85: 7; Apr 85: 7; May 85: 7

E. Subject Index

Abelson
  Sep 82: 1, 7; Nov 82: 3; Dec 82: 1, 8; Jan 83: 8; May 83: 10; Dec 83: 6; Mar 84: 7, 14; Sep 84: 8; Mar 85: 1
adventure games
  Dec 83: 5; Jan 84: 11; Apr 84: 9; May 84: 7
Allen, J.
  Mar 84: 7
ASCII
  Feb 84: 9
ASK (Mattel Logo)
  Oct 83: 10
Atari Logo (using SETSP, TOOT)
  Sep 83: 1
Bearden, D.
  Jan 83: 1; May 83: 10; Oct 83: 12; Feb 84: 4, 5; Oct 84: 1; Apr 85: 12, 13
Bitter, G.
  May 83: 10
Bowen, B.
  May 83: 10
Burnett, J.
  Mar 83: 1; Nov 83: 5
CHAR
  Dec 83: 10; Feb 84: 9;
commands and operations
  Sep 84: 11 (differences); Apr 85: 14 (reserved characters)
Commodore Logo
  Nov 83: 9; Dec 84: 8; Mar 85: 1X
concrete materials
  Nov 83: 1; Feb 84: 1; Sep 84: 13
content areas and Logo (integration into the curriculum)
  Jan 83: 5; Feb 83: 1; May 83: 1; Dec 83: 3; Jan 84: 4; Sep 84: 3
  art-- Nov 84: 1
  geometry-- Sep 82: 1; Oct 82: 1,7; Jan 84: 7; Feb 84: 5; Mar 84: 5; Nov 84: 14; Jan 85: 1
  language arts-- Apr 83: 5; May 83: 5; Jan 84: 11; see "Listful Thinking," Bull & Cochran, Volume 3
  physics-- Sep 82: 7; Jan 84: 7; May 84: 11; Sep 84: 2X; Nov 84: 7; Feb 85: 2X
Cron, M.
  Feb 84: 12
COUNT
  Sep 84: 7; Nov 84: 7
Davis, J.
  May 83: 10
debugging
  Dec 82: 1
diSessa, A.
  Sep 82: 1; Nov 82: 3; Apr 84: 8; Dec 84: 16

dissertations dealing with Logo
  see "Logo Disserts," Barbara Elias, Volume 3
DOS
  Sep 84: 1
educational benefits
  Oct 82: 1; Dec 82: 3
Evans, H.
  Apr 85: 11
extensibility
  Sep 84: 7
Family Opportunity Act
  Sep 82: 6
Ferres, G.
  Mar 85: 11
floor turtle
  Sep 82: 2; Mar 83: 8; May 83: 1, 5; May 85: 2X
Folk, M.
  Jan 85: 8
fractals
  Oct 82: 7
Green, C.
  May 84: 13; May 85: 12
Harper, D. O.
  May 85: 4
Horn, M.
  May 83: 10; Jan 85: 12; Mar 85: 12
IF
  Mar 85: 15; Apr 85: 15; May 85: 1X
INSPECT
  Nov 84: 7; Dec 84: 9; Feb 85: 9
INSTANT
  May 83: 1; Dec 83: 6; Jan 85: 14
ITEM
  Sep 84: 7
Lawler, R.
  Oct 82:1
LIST
  Jan 84: 11;
list operators (FIRST, BUTFIRST, LAST, etc.)
  see "Listful Thinking," Bull & Cochran, Volume 3
  Apr 83: 5; May 83: 5; Nov 83: 6; Dec 83: 10; Oct 84: 13; Mar 85: 1
list processing
  see "Listful Thinking," Bull & Cochran, Volume 3
  Feb 83: 2; Oct 83: 6; Nov 83: 6; Dec 83: 5; Jan 84: 11; Feb 85: 1
Logo (discussion/ issues)
  Sep 82: 5 [Bull]; Oct 82: 1 [Towberman, parent];
  Oct 82: 5 [Bull]; Nov 82: 5 [Bull]; Dec 82: 5 [Bull]
  Jan 83: 5 [Bull]; Jan 85: 6 [Wigley]
Logo environment
  Dec 82: 3; Mar 83: 5; Apr 83: 7; Dec 83: 3; Jan 84: 4; Sep 84: 7;
  Sep 84: 3
Logo in Japan (see also Weintraub columns)

Jan 84: 14
Logo User's Groups (see Logo around the world)
  Boston-- Mar 84: 4; Apr 84: 8
  directory-- Mar 85: 13
  France-- Apr 85: 14
  Japan-- Nov 84: 16
  Michigan-- Nov 84: 14
  Portland, Oregon-- Sep 84: 2X
  Special Education-- Mar 85: 16
Lohrman, L.
  Dec 83: 16
Mandelbrot
  Sep 82: 8 (fractals)
Martin, K.
  Jan 83: 1; Oct 83: 12; Oct 84: 1, 2; Apr 85: 12, 13
McDougall, A.
  Apr 84: 15
MEMBER?, MEMBERP
  Apr 83: 5; Dec 84: 9
microworlds
  Dec 83: 2; Apr 84: 3
Mindstorms
  Sep 82: 1; Oct 82: 6; Apr 83: 8; Dec 83: 14; Mar 84: 7
modularity
  Jan 83: 5; Sep 83: 3; Dec 83: 3; Jan 84: 7; Apr 84: 1
Moscovitch, J.
  Nov 84: 10
Moursund, D.
  Jan 84.2; Feb 84: 14
multiple turtles (see also-- sprites)
  Sep 83: 1; Oct 83: 10;
Muller, J.
  Jan 83: 1; Apr 83: 1; Feb 85: 6
NLX ABC's
  Apr 83: 1, 6; Dec 83: 5; Sep 84: 2X; Mar 85: 1X
Noss, R.
  Feb 84: 13
NOVA's "Talking Turtle" documentary
  Dec 83: 14
NUMBER?, NUMBERP
  Mar 85: 14
OUTPUT
  Nov 83: 6; Jan 84: 7; Apr 84: 9; May 84: 3; Sep 84: 7; Oct 84: 7;
  Nov 84: 7, 11; Dec 84: 1
Papert
  Sep 82: 1, 5; Oct 82: 6; Apr 83: 8; Dec 83: 3, 14; Feb 84: 13;
  Mar 84: 3, 4, 7; Apr 84: 13; May 84: 1; Sep 84: 14; Dec 84: 16;
  Jan 85: 4; Mar 85: 13; May 85: 2, 5
Pascal
  Jan 83: 8; (newsletter) Feb 85: 2
PAUSE
  Apr 85: 1, 6 (PAWS for those versions without PAUSE)

Pea, R.
Mar 84: 4
peripherals
Nov 82: 5; Mar 84: 11; Sep 84: 9
physics
Jan 84: 7; Apr 84: 8, 14; May 84: 11
Piaget
Sep 82: 1; Oct 82: 3; Oct 84: 15; Jan 85: 3, 8; Feb 85: 16
PICK
Sep 84: 7; Oct 84: 7; Nov 84: 7; Feb 85: 9
Polya
Dec 82: 1; Apr 84: 16; Oct 84: 12
problem solving
Dec 82: 1; Jan 83: 3, 5; Dec 83: 2, 3; Sep 84: 10; Oct 84: 12; Feb 85: 12; Apr 85: 6
RANDOM, RANDOMIZE
Mar 83: 3; Dec 84: 8; Oct 84: 15
READLIST, REQUEST
Sep 83: 6; Mar 85: 14
READWORD
Dec 84: 9
recursion
Feb 84: 5; May 85: 13
REPEAT
Oct 82: 3; Nov 82: 3; Jan 83: 3
Rieber, L.
Oct 84: 14; Nov 84: 14
Seidman, R.
Feb 85: 12
self-esteem/self-confidence
Oct 82: 1; May 84: 1
SENTENCE
Sep 84: 7; Oct 84: 13
Soergel, D. & L.
Oct 83: 9
special education (handicapped)
Nov 82: 5; Dec 83: 6; May 84: 1; Mar 85: 16, 2X; Apr 85: 12
sprites
Nov 82: 1; Mar 83: 1; May 83: 1; Sep 83: 1; Oct 83: 10; Nov 83: 9, 10; Mar 84: 4; Apr 84: 6; Jan 85: 6; Feb 85: 6
Statz, J.
Oct 84: 13
STEP
Apr 85: 1
teacher's role
Nov 82: 2; Dec 82: 3; Jan 83: 3; Oct 83: 1; Dec 83: 3; Jan 84: 4; Feb 84: 12; Sep 84: 12; May 85: 1; Feb 85: 8; May 85: 6
teacher training
Dec 82: 3; Mar 84: 1; Mar 85: 11
teaching materials
see "Teacher to Teacher," Volume 3
teaching tips

animated signs-- Oct 84: 3
arithmetic operations-- Oct 83: 3
curves-- Apr 84: 14; Jan 85: 1, 3
debugging-- Nov 82: 3; Dec 82: 1; Mar 85: 14; Apr 85: 1
degrees (understanding) Oct 84: 6
distance and direction-- Sep 82: 2; Nov 82: 3; Jan 83: 3
drawing frames-- Sep 83: 3
DOS-- Sep 84: 1
editting-- Mar 85: 7
elem-aged students (also kindergarten)-- Sep 82: 1;
  Dec 82: 1, 3; Jan 83: 1; May 83: 1; Dec 83: 1, 5; Feb 83: 1;
  Jan 84: 1; Apr 84: 1; Nov 84: 1
fractals-- Oct 82: 7
fractions-- May 83: 3
functions-- Dec 84: 1
games-- Sep 82: 2; May 84: 3; Nov 84: 3
geometry-- Sep 82: 1; Jan 84: 7; Feb 84: 5; Mar 84: 5; Apr 84: 3;
  May 84: 3; Nov 84: 14; Jan 85: 1, 3
gifted students-- Feb 83: 1
grammar and the computer-- Apr 83: 5; May 83: 5
graphics-- May 83: 8, 11; Nov 84: 1
grids-- Feb 84: 1; Apr 84: 9; May 84: 7
holiday ideas-- Oct 84: 7; Dec 84: 9 see "TurtleTips," Lanyi & Toth,
  Volume 3
keyboarding-- Nov 84: 6
language arts-- Jan 84: 11; May 84: 9; see "Listful Thinking,"
  Bull & Cochran, Volume 3
list processing (word projects)-- Feb 83: 3; May 84: 9;
  Sep 84: 7; Oct 84: 7; Nov 84: 7; Feb 85: 1; Mar 85: 1
  see "Listful Thinking," Bull & Cochran, Volume 3
math drills (creating)-- Mar 83: 3
mazes-- Oct 84: 7
modeling-- Oct 83: 1
multiple turtles-- Nov 82: 1; Mar 83: 1; May 83: 1; Oct 83: 10
NLX ABCs-- Apr 83: 1; Dec 83: 5
numbers-- Oct 83: 3, 6; Nov 84: 3; Dec 84: 3;
off-computer activities-- Sep 84: 13; Oct 84: 1;
OUTPUT-- Apr 83: 5; May 84: 3; Nov 84: 14; Dec 84: 1
overlays-- Nov 82: 1; Dec 82: 8
pairs (of students)-- Feb 83: 1; Jan 85: 7; May 85: 1
patterns, repetition-- Sep 82: 7; Oct 82: 3, 7; Nov 82: 3;
  Dec 82: 7; Apr 83: 1; May 83: 9
physics-- Nov 82: 7; Apr 84: 14; May 84: 11
preschoolers-- Sep 82: 2; Nov 83: 1
procedure (concept)-- Jan 83: 3; Jan 84: 4; Mar 85: 7
PRINT (and CURSOR)-- Feb 83: 3, 5
RANDOM-- Mar 83: 3 (probability); Dec 84: 8; Oct 84: 3
scheduling, time management-- Feb 83: 1; May 85: 1; Sep 84: 3;
  Jan 85: 7
scrunch commands-- Apr 84: 3
slowing down the turtle-- Sep 83: 7
surveys-- Feb 85: 3; Mar 85: 3; Apr 85: 3; May 85: 3

  symmetry-- Nov 83: 3
  talking with Logo-- May 84: 1
  text-- Feb 83: 3, 5; Mar 84: 9
  trigonometry (sine, cosine)-- Jan 83: 7; May 84: 11
  writing project-- Feb 83: 8
Technology Education Act of 1982
  Sep 82: 6
Thornburg, D.
  Dec 84: 12
tools/useful ideas
  BLOADing pictures-- Sep 84: 1
  calculating percentages-- Sep 83: 6
  cleaning out the workspace-- Oct 83: 5
  deactivating auto-repeat key-- Jan 85: 14
  debugging-- Apr 85: 1
  Delta Drawing and Logo-- Jan 85: 2X
  DOS--Sep 84: 1
  erasing a line-- Feb 83: 5; Nov 84: 2X
  exploring poetry, words-- Sep 84: 7; Oct 84: 7; Nov 84: 7
  generating story starters-- Nov 84: 7
  grade calculator, averager-- May 83: 3; Nov 83: 6
  lists-- see "Listful Thinking," Bull & Cochran, Volume 3
  Logo and ABC's-- Dec 82: 2; Apr 83: 1
  Logo and drive 2-- Oct 83: 3
  parallel printer and Apple Logo II-- May 85: 4X
  PAWS for versions without PAUSE-- Apr 85: 6
  plotters-- Mar 84: 11
  printing inverse letters-- Nov 84: 2X
  printout of LCSI procedures-- Oct 83: 9
  report generator-- Dec 83: 10
  set the turtle to a random heading-- Sep 82: 4
  storing word lists, data structures-- May 83: 3; Feb 85: 3;
    Mar 85: 14
  surveys-- Feb 85: 3; Mar 83: 3; Apr 85: 3; May 85: 3
  talking with Logo-- May 84: 1
  Terrapin utilities-- Feb 85: 6
  text--Feb 83: 5; Mar 84: 9
  WRITE- Feb 85: 9
Total Turtle Trip Theorem
  Nov 82: 3; Jan 83: 3
TRACE
  Apr 85: 1
turtle song
  Nov 85: 15
variables
  Jan 83: 3; Feb 83: 3; Apr 83: 3; Oct 84: 3; Dec 84: 14; Feb 85: 14
Visi-Calc
  Dec 82: 5
WAIT
  Feb 83: 5
Wappler, R.
  Oct 83: 5; Feb 85: 14

Watt, D.
Jan 83: 8; Oct 83: 9; Mar 84: 4; Sep 84: 4; Nov 84: 5
Watt, M.
Oct 82: 1
Weir, S.
Dec 84: 6
Young, L.
Sep 84: 10
YPLA (Young Peoples' Logo Association)
Jan 83: 1; Oct 83: 3; Mar 84: 14; Nov 84: 1X; May 85: 14